

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DEL PERÚ

PROGRAMACIÓN DEL SISTEMA ELECTRÓNICO DE UN ROBOT
MANIPULADOR DE CINCO GRADOS DE LIBERTAD

Tesis para optar el Título de **Ingeniero Electrónico**, que presenta el bachiller:

José Sergio Chávez Méndez

ASESOR: Willy Carrera Soria

Lima, Diciembre de 2014

Resumen

En el presente trabajo de tesis se desarrolla la programación necesaria para manipular los cinco grados de libertad de un robot manipulador serial angular, creado por la Maestría de Ingeniería Mecatrónica de la Pontificia Universidad Católica del Perú. Dicha programación se encuentra parametrizada por el hardware heredado, el cual estará detallado en el trabajo de tesis titulado: "Diseño e implementación del hardware del sistema electrónico de un robot manipulador de cinco grados de libertad".

Se requiere, principalmente, que el usuario consiga manipular de forma directa la posición angular a la que se solicite que cada articulación llegue. Esto debe realizarse a partir de la implementación de un sistema de control retroalimentado.

Para tal fin se programaron la interfaz de usuario y el software de los microcontroladores utilizados bajo el modelo Maestro-Eslavos. La interfaz recibe los ángulos solicitados por el usuario y los envía al microcontrolador maestro que a su vez envía los datos a cada una de las articulaciones para las cuales se les solicitó movimiento; de esta forma, los microcontroladores esclavos realizan la ejecución del algoritmo de control necesario para llegar a la posición solicitada.

Se implementaron cinco controladores PD monoarticulares de posición angular para cada grado de libertad mediante la emulación discreta de controladores continuos; esto se consiguió a partir de la obtención del modelo matemático de cada articulación a controlar según ensayos realizados.

En el capítulo uno, se describe qué es y el por qué es importante un robot manipulador; asimismo, se describen algunos tipos de robots manipuladores. En el capítulo dos, se detalla los bloques funcionales que posee la electrónica de un robot manipulador y como estos son encarados. En el capítulo tres, se aborda el diseño y la implementación de la programación del sistema. Finalmente, en el capítulo cuatro, se describen las pruebas realizadas y se muestran los resultados del sistema implementado.

Dedicatoria

A mi padres Liliana y Ángel por darme durante todos estos años un incommensurable apoyo e incondicional cariño, además de ser mis grandes modelos a seguir

A mis queridos hermanos Karol, Piero y José Ángel por todas las alegrías que hemos compartido.

A mi abuelo Papito peyo que desde el cielo sé que nos cuida.

A mis abuelas Julia y María Julia por todo el cariño y alegrías recibidas.

A mis asesor Willy Carrera por la confianza puesta en mí y estar siempre presto a ayudarme.

A mis grandes amigos Luis Rodriguez, Misely Ruiz, Arthur Ramos, Miguel Fabián, Sergio Shimabukuro, Stefanni Corrales, Matías Quintana, Miguel Bazán, Frank Salvador, José Miguel Molina, Sammy Cerida y Guillermo Garayar.

A mi querida Asociación de Estudiantes de Electrónica (AEE) y las grandes personas que conocí ahí. A Héctor Rivas, Renzo Garaycochea, Gabriela Torres, Anthony Gonzales, Juan Huayta, Hideki Masuda, Patricia Rodriguez, Stefano Romero y demás compañeros de la AEE por los grandes y únicos momentos compartidos.

A mis compañeros del grupo de microelectrónica (GuE) por el gran apoyo durante el desarrollo de esta tesis.

¡Gracias!

Índice general

Resumen	I
Introducción	VI
1. Robots manipuladores	1
1.1. Robots	1
1.2. Robots manipuladores	1
1.2.1. Descripción	1
1.2.2. Tipos de robots manipuladores	2
2. Etapas en la electrónica de un robot manipulador	5
2.1. Planteamiento	5
2.2. Etapas	5
2.2.1. Actuador	5
2.2.2. Excitador	6
2.2.3. Sensor y acondicionamiento	6
2.2.4. Controlador	7
2.2.5. Algoritmo de control	8
2.2.6. Interfaz de usuario	8
2.3. Síntesis	8
2.4. Objetivos	9
3. Diseño e Implementación de los controles electrónicos e interfaz de usuario	10
3.1. Alcances	10
3.2. Diagrama de bloques	11
3.3. Hardware	13
3.3.1. Planta	13
3.3.2. Actuadores	14
3.3.3. Excitadores	14
3.3.4. Sensores	15
3.3.5. Controladores	16
3.4. Software	21

3.4.1. Interfaz de usuario	21
3.4.2. Software del microcontrolador Maestro	25
3.4.3. Software de los microcontroladores Esclavos	27
4. Ensayos	46
4.1. Interfaz de usuario	47
4.2. Microcontrolador maestro	48
4.3. Microcontroladores esclavos	48
Conclusiones	52
Recomendaciones	53



Índice de figuras

1.1. Robot manipulador cartesiano [10]	2
1.2. Robot manipulador cilíndrico [13]	3
1.3. Robot esférico <i>Unimate</i> [17]	3
1.4. Robot manipulador angular [14]	4
1.5. Robot S.C.A.R.A. [6]	4
1.6. Estructura básica de los cuatro primeros tipos de robots manipuladores mencionados. [8]	4
3.1. Diagrama de bloques	12
3.2. Robot manipulador - Articulación 1 y 2 ensambladas	13
3.3. Tarjeta del excitador VNH2SP30 MD03A [5]	15
3.4. Sensor de final de carrera	16
3.5. Conversor USB a Serial TTL	17
3.6. Diagrama esquemático de la tarjeta Maestro	18
3.7. Diagrama esquemático de las tarjetas Esclavas	20
3.8. Etapa de comunicación serial de la interfaz	21
3.9. Diagrama de flujo de los eventos asociados a la Comunicación Serial	22
3.10. Diagrama de flujo del envío de trama vía serial	23
3.11. Etapa de cinemática directa	24
3.12. Diagrama de flujo principal del microcontrolador Maestro	25
3.13. Diagrama de flujo del envío de ángulos a los controladores esclavos	27
3.14. Diagrama de flujo principal de los microcontroladores Esclavo	28
3.15. Diagrama de la subrutina encargada del algoritmo de movimiento	30
3.16. Diagrama de bloques de control	31
3.17. Diagrama del modelo de un motor DC [4]	31
3.18. Circuito utilizado para la medición de parámetros	32
3.19. Desfase de ondas en el ensayo para el actuador M1	33
3.20. Diagrama de bloques de la simulación del sistema de control de la articulación	39
3.21. Respuesta a un pulso. Articulación 1	39
3.22. Respuesta a un pulso. Articulación 2	40
3.23. Respuesta a un pulso. Articulación 3	41

3.24. Respuesta a un pulso. Articulación 4	41
3.25. Respuesta a un pulso. Articulación 5	42
3.26. Diagrama de flujo del algoritmo de control	43
3.27. Rizado en estado estacionario	44
4.1. Interfaz de usuario	47
4.2. Interfaz de usuario	47
4.3. Trama serial visualizada	48
4.4. Trama I^2C para dos esclavos	48
4.5. Posición angular de la articulación M1	49
4.6. Respuesta de control de posición - Articulación 1	49
4.7. Respuesta en velocidad de M1	50
4.8. Respuesta de la simulación	50
4.9. Duración de la subrutina de control	51



Índice de cuadros

3.1. Ángulos de giro permitidos	10
3.2. Distancia entre los orígenes de los ejes de los motores	13
3.3. Características de los actuadores. Fuente: Hojas de datos	14
3.4. Harmonic Drives y las constantes de reducción.	14
3.5. Características del excitador. Fuente: Elaboración propia	15
3.6. Características de los <i>encoders</i> recibidos	16
3.7. Características del microcontrolador <i>ATmega8A</i>	17
3.8. Ejemplo de trama enviada por serial	24
3.9. Parámetros de Denavit-Hartenberg	24
3.10. Posición inicial de los actuadores	29
3.11. Cálculo de resistencias e inductancias de armadura	34
3.12. Constantes de torque y voltaje	34
3.13. Cálculo de rozamientos viscosos	35
3.14. Datos utilizados para el cálculo del momento de inercia	37
3.15. Cálculo del momento de inercia (J_a)	38
3.16. Límite inferior de saturación de voltaje	45

Introducción

A lo largo de la historia, la robótica es un campo que ha evolucionado dramáticamente en el ámbito internacional, sobre el cual se han desarrollado e implementado inconmensurables maneras de afrontar los problemas que pueden suponer el desarrollo de un robot [15]. Lamentablemente, en nuestro país este no es el caso; el área no ha sido investigada de la misma forma en la que un país desarrollado lo hace, pues no se cuenta con la difusión y el apoyo necesario.

A fin de suplir esta carencia, nuestra casa de estudios busca incentivar la investigación y el desarrollo en esta área; razón por la que, mediante la Maestría de Ingeniería Mecatrónica, se desarrolló e implementó la mecánica completa de un robot manipulador de cinco grados de libertad; esto comprende: el forjado de las articulaciones mecánicas, la adquisición de sensores, actuadores y reductores, y, finalmente, el ensamblado del sistema mecánico.

El presente trabajo de tesis propone una forma de abordar el problema de la lógica de funcionamiento, necesaria para la manipulación de movimientos simples de las articulaciones del robot de cinco grados de libertad mencionado. Esto se realiza con el objetivo de sentar bases para futuros trabajos de investigación que continúen con el presente proyecto, pudiendo abordar los temas que continúan, los cuales pueden ser el control de la cinemática inversa y la generación de trayectorias.

Capítulo 1

Robots manipuladores

1.1. Robots

Los robots son entidades electromecánicas capaces de interactuar con su entorno, mediante una lógica previamente programada en el computador interno que estos poseen a fin de realizar tareas que se encontraban reservadas solo a personas [1]. Estos son mecanismos físicos que pueden sentir y manipular el entorno en el que se encuentran, capaces de moverse y efectuar fuerza. Se pueden encontrar en diversos ámbitos para una amplia cantidad de aplicaciones. Dentro de los robots electromecánicos se pueden encontrar los robots manipuladores.

1.2. Robots manipuladores

1.2.1. Descripción

El estándar *ISO 8373* [16] define a los robots manipuladores como una máquina que consiste usualmente en una serie de segmentos, capaces de realizar movimiento relativo entre uno y otro segmento con el propósito de coger o mover objetos con varios grados de libertad. Por tanto estos poseen una gran versatilidad en sus aplicaciones y, en consecuencia, son ampliamente utilizados en distintos campos, algunas de estos son: en la industria, para automatizar tareas repetitivas; en la medicina, para realizar tareas críticas de alta precisión; y, también, en el ámbito científico, para la investigación de su diseño, programación e implementación.

Uno de los campos en el que más popularidad poseen los robots manipuladores es en el de la industria [12]. En ella, son necesarios para automatizar ciertas partes del proceso de producción, que de otra forma hubieran requerido personal adicional o que, simplemente, no se hubieran podido realizar; dentro de las capacidades de estos podemos encontrar las siguientes: manipulación de grandes cargas, las cuales de otra manera no hubieran sido posible de manipular debido al gran esfuerzo físico que se requeriría; realizar tareas de gran precisión pues es posible llegar a tener un error ínfimo e imperceptible, que de otra forma estaría supeditado a la pericia del operario; y, por último, realizar tareas de gran velocidad, que de no ser así no se produciría de la forma en que se requiere.

Por dichas razones expuestas, se ha desarrollado el tema extensamente de forma comercial y podemos encontrar en el mercado distintos modelos que cumplen con las especificaciones requeridas según la tarea para la cual se les requiere.

Por otro lado, los robots manipuladores son también muy estudiados en el ámbito científico. En él se busca desarrollar -y también aprender- las distintas formas que hay para afrontar los problemas que suponen el diseño y la implementación de un robot manipulador.

1.2.2. Tipos de robots manipuladores

Dentro de los tipos más comunes de robots manipuladores podemos encontrar los siguientes:

1.2.2.1. Robot manipulador cartesiano

Este tipo de robot manipulador posee tres articulaciones prismáticas (Figura 1.6). Permite lograr una gran precisión con cargas pequeñas; sin embargo, es posible encontrar modelos que permiten manipular cargas voluminosas. Dado que la manipulación del robot se realiza ingresando las coordenadas cartesianas (x , y , z) el control es más simple. No obstante, tiene un difícil acceso a ciertos ángulos y poco al alcance [24].

En la Figura 1.1 se aprecia un manipulador cartesiano comercial de la marca *AVM*. Sus actuadores son neumáticos y puede llegar a manipular un carga de hasta 20Kg [10].



Figura 1.1: Robot manipulador cartesiano [10]

1.2.2.2. Robot manipulador cilíndrico

Este robot posee una estructura similar al manipulador cartesiano; sin embargo, en vez de una articulación prismática en la base, este es capaz de rotar sobre la misma (Figura 1.6). Es muy utilizado para aplicaciones del tipo “coger y dejar” [24]. En la Figura 1.2 se aprecia el manipulador cilíndrico *R19* de la marca *ST Robotics*, el cual soporta hasta 2.0Kg de carga. Este robot es muy utilizado en laboratorios para la manipulación de tubos de ensayo [13].



Figura 1.2: Robot manipulador cilíndrico [13]

1.2.2.3. Robot manipulador polar o esférica

Este robot posee dos articulaciones rotacionales y una prismática (Figura 1.6). Tiene un buen alcance. No obstante, está prácticamente desaparecido del mercado [24]. En la Figura 1.3 se muestra una versión moderna del robot manipulador esférico *Unimate*, primer robot industrial en la historia [2].

Figura 1.3: Robot esférico *Unimate* [17]

1.2.2.4. Robot manipulador angular

Posee al menos tres articulaciones del tipo rotacional, asemejándose mucho a un brazo humano (Figura 1.6), lo cual otorga un mayor campo de acción. Este es el manipulador más utilizado en aplicaciones industriales [24]. En la Figura 1.4 se aprecia un manipulador angular *R2000iA* de la marca *Fanuc*, el cual es capaz de manipular cargas de 200Kg y posee un alcance máximo de 2.92m, es muy utilizado en la industria automotriz [14].



Figura 1.4: Robot manipulador angular [14]

1.2.2.5. Robot S.C.A.R.A.

Posee tres articulaciones rotacionales de ejes paralelos y una articulación prismática y, al igual que los manipuladores cilíndricos, son muy utilizados para aplicaciones del tipo “coger y dejar”. Son construidos especialmente para tareas de montaje sobre un plano [24]. En la Figura 1.5 se muestra el robot S.C.A.R.A. *G20* de la marca *Epson*, puede cargar hasta 20Kg y posee hasta un metro de alcance.



Figura 1.5: Robot S.C.A.R.A. [6]

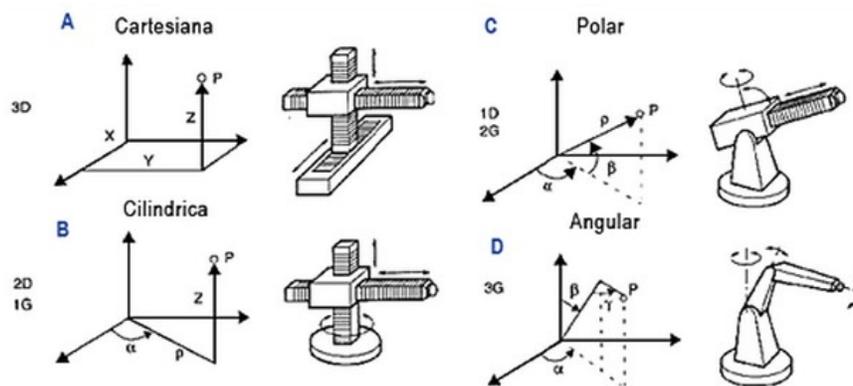


Figura 1.6: Estructura básica de los cuatro primeros tipos de robots manipuladores mencionados. [8]

Capítulo 2

Etapas en la electrónica de un robot manipulador

2.1. Planteamiento

Al ser una gran herramienta para el aprendizaje y desarrollo, la Maestría de Mecatrónica, Pontificia Universidad Católica del Perú, ha llevado acabo la implementación de la estructura mecánica de un robot manipulador de cinco grados de libertad; para el cual se requiere diseñar el sistema electrónico que posea la capacidad de llevar eficazmente cada grado de libertad del robot a la posición angular solicitada. Para lograr una solución eficaz se realiza un análisis de las etapas que existen en la electrónica de un robot manipulador.

2.2. Etapas

Las etapas de funcionamiento de la electrónica de un robot manipulador se puede dividir en los siguientes bloques funcionales: Actuador, Sensor, Controlador, Algoritmo de control, Excitador e Interfaz de usuario.

2.2.1. Actuador

Los actuadores son la parte del robot sobre la cual se ejecuta directamente la acción de control y se encargan de efectuar el trabajo necesario para controlar el robot manipulador. A continuación se describe los tipos de actuadores más usados y sus características [20].

- Actuadores neumáticos: Los actuadores neumáticos no poseen una gran precisión y, generalmente, son usados cuando se requiere tener solo dos posiciones (todo o nada).
- Actuadores hidráulicos: A diferencia de los neumáticos, en los actuadores hidráulicos sí es posible realizar un control continuo y, además, pueden poseer una gran precisión y un elevado par motor. Sin embargo, dadas las elevadas presiones con las que trabajan, se requiere un

sistema que permita proteja el robot ante fugas, elevadas temperaturas y eliminación del aire.

- Actuadores eléctricos: Son los actuadores más utilizados en los robots manipuladores ya que se pueden controlar de forma más sencilla y eficaz. Dentro de los actuadores eléctricos más usados tenemos:
 - Motor DC: Por su simplicidad, son el tipo de actuadores eléctricos más usados en los robots manipuladores. Dado que poseen un comportamiento muy lineal, permiten realizar un control eficaz y continuo de su velocidad y posición. Adicionalmente, se les puede acoplar fácilmente al eje un *encoder* para el sensado de su velocidad y posición.
 - Motor paso a paso: Son muy poco usados puesto a que no poseen un adecuado par motor en comparación con los motores DC. Se utiliza normalmente cuando no se requiere una gran torque, como en el giro de pinzas o en robots pequeños.
 - Motor AC: Inicialmente no eran utilizados en robótica dada la complejidad de su control; no obstante, dadas las mejoras en la tecnología del diseño de estos motores, ya es posible encontrarlos en campo siendo controlador con alta precisión y velocidad.

2.2.2. Excitador

Usualmente, los actuadores eléctricos poseen un consumo de corriente elevado; razón por la que siempre es pertinente evaluar la adquisición de un excitador, que permita manipular el actuador a petición del controlador a usar. Estos pueden ser implementados en base a circuitos integrados o componentes discretos y deben permitir la manipulación del sentido de giro y la velocidad.

Es posible encontrar excitadores comerciales; estos pueden venir con funciones adicionales, como sensado de corriente, protección ante cortocircuito, protección por sobrevoltaje, etc; no obstante, el precio es mayor en comparación a un excitador diseñado en base a componentes discretos.

2.2.3. Sensor y acondicionamiento

El sensado de la posición angular de un robot manipulador se puede realizar de distintas formas, las formas más comunes son las siguientes [24]:

- Potenciómetro: Para determinar la posición angular es posible anexar un potenciómetro lineal de precisión al eje del motor y, en base al sensado de su resistencia eléctrica, se obtiene la posición angular. Estos pueden presentar errores debido a que son muy susceptibles al desgaste por la fricción.
- *Encoder* incremental: El *encoder* incremental entrega por el canal A una cierta cantidad de pulsos proporcional a la cantidad de ángulos de giro del eje del motor. Para saber la cantidad en ángulos sexagesimales que ha girado dicho eje, basta con dividir la cantidad de pulsos que se han obtenido con la resolución del *encoder* y multiplicarlo por 360° . Los *encoders* poseen

también un canal B que es utilizado para determinar el sentido de giro del motor, pues este entrega el mismo tren de pulsos que el canal A pero desfasado 90° o -90° , según el sentido de giro.

- *Encoder* absoluto: Similar al *encoder* incremental, a diferencia de que el *encoder* absoluto asigna a cada posición angular un código (usualmente Gray) que puede ser leído por el sistema para tener la posición absoluta en cualquier momento en que esta sea requerida. La codificación y el grado codificado depende de la resolución del *encoder*.
- Resolver: Es un dispositivo analógico al cual se le entrega una señal sinusoidal portadora y este devuelve dos señales sinusoidales desfasadas 90° , proporcionales al ángulo de giro. De esta forma se puede saber la posición angular del eje y el sentido de giro del mismo. El principal problema en la utilización del resolver, es que se necesita circuitería adicional para acondicionar la señal que este entrega.

No obstante, un robot manipulador podría no solo tener sensores de posición, sino también es posible encontrar sensores de fuerza, presencia y proximidad, de navegación y demás sensores que le permiten interactuar con el entorno que manipulan.

2.2.4. Controlador

La implementación del servo-controlador y de la lógica de funcionamiento se realiza generalmente en un computador embebido o en un computador personal (PC). La principal ventaja de implementarlo en uno embebido radica en el bajo costo que su realización implica; sin embargo, hay que tener consideraciones que no se da con la implementación en PC; estas son:

- La cantidad de microcontroladores necesarios para los distintos grados de libertad.
- Se debe diseñar la arquitectura del arreglo de microcontroladores a usar, siendo usual tener un maestro y varios esclavos.
- La cantidad pines de salida o entrada es crítico para la elección del microcontrolador
- La velocidad del reloj del microcontrolador tiene que ser apropiada para poder para poder hacer el correcto seguimiento al sensor y para poder ejecutar eficazmente el algoritmo de control.
- Usar un microcontrolador implica circuitería extra para su funcionamiento y lo cual se traduce en costo y espacio adicional.

Por otro lado, en la implementación en PC, no se necesita hacer énfasis en las consideraciones mencionadas; sin embargo, el costo de implementación es mucho mayor, puesto que usualmente se opta por adquirir una tarjeta de adquisición de datos (DAQ) que funciona como interfaz entre la PC y el sistema que se requiere controlar; dicha tarjeta de adquisición debe poseer la cantidad de puertos de E/S necesarios para seguir los sensores y manipular los excitadores. No obstante, es posible encontrar otros dispositivos que sirvan de interfaz (p.e: microcontroladores).

2.2.5. Algoritmo de control

El control de la posición de un motor se puede realizar bajo diversos métodos, estos se pueden dividir en dos grandes ramas: control clásico y control moderno.

En el control clásico se encuentran implementaciones mediante el uso del algoritmo PID y, más eficientemente, controladores de espacio-estado. Sin embargo, debido a que los robots manipuladores son sistemas no lineales, es común el uso técnicas de control moderno; como pueden ser control difuso o control por redes neuronales, estos tipos de controladores permiten trabajar de forma más eficiente con sistemas no lineales; no obstante, implican una mayor complejidad de diseño y/o implementación.

2.2.6. Interfaz de usuario

La interfaz de usuario permite que se pueda controlar el robot manipulador de forma práctica y directa, para esto existen diversas maneras; no obstante, generalmente se desarrolla un software en un computador personal dada la flexibilidad que este posee.

2.3. Síntesis

Dentro de las principales consideraciones al implementar un robot manipulador tenemos: la elección del hardware, el diseño del controlador y la interfaz de usuario. Estos se han afrontado de distintas formas.

Para el diseño de un robot manipulador es importante elegir el hardware según el propósito de diseño del robot; en esta parte se encuentran implicadas la elección del actuador y la elección del sensor.

Como se mencionó, los actuadores elegidos usualmente son motores DC, debido a que su simplicidad hace que sea mucho más fácil aplicar técnicas de control, su bajo costo es también un factor determinante en su elección; sin embargo, cuando no se requiere un par motor elevado se puede considerar motores a pasos. Asimismo, para manipular la potencia requerida por los actuadores y separarla de la parte lógica, usualmente se diseñan excitadores bajo dichos requerimientos.

Los sensores más utilizados son encoders incrementales dado que son muy precisos y a que partir de estos no solo se puede conocer la posición, sino también la velocidad, sentido de giro y la cantidad de vueltas completas dadas. Asimismo, los potenciómetros son también muy utilizados para sensar la posición; sin embargo, estos no son tan precisos.

Por otro lado, el diseño del algoritmo de control es uno de los temas más investigados en la electrónica de los robots manipuladores, dado que estos determinan en última instancia la precisión y el tiempo de respuesta del sistema completo. Pueden basarse en el control moderno o en el control clásico, y se eligen dependiendo de las características que primen en la tarea a realizar ya sea torque, velocidad y/o precisión. Uno de los más usado es el controlador PID debido a que su implementación es más sencilla [22], usualmente posee variaciones como compensación por gravedad o de forma adaptiva.

Finalmente, la interfaz de usuario se implementa por lo general en un computador personal; sin embargo, también es posible encontrar sistemas embebidos que cumplan con esta función [23].

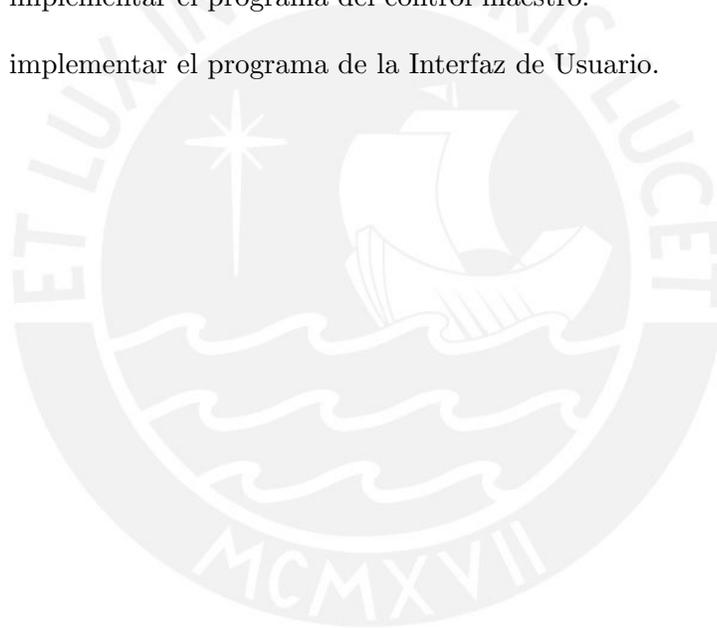
2.4. Objetivos

Objetivo general:

- Diseñar e implementar la programación del sistema electrónico que permita manipular la posición de un brazo robot articulado de cinco grados de libertad.

Objetivos específicos:

- Diseñar e implementar el programa de cada uno de los 5 controles esclavos.
- Diseñar e implementar el programa del control maestro.
- Diseñar e implementar el programa de la Interfaz de Usuario.



Capítulo 3

Diseño e Implementación de los controles electrónicos e interfaz de usuario

3.1. Alcances

Se describe de forma detallada el diseño de los programas de control propios de los controladores Maestro-Eslavos y de la Interfaz de Usuario, que permita la manipulación de la posición angular de cada una de las articulaciones de un robot manipulador de cinco grados de libertad. El sistema debe recibir información de los sensores durante el movimiento de las articulaciones para conseguir reducir el error al mínimo posible y, de esta forma, llegar a la posición solicitada por el usuario. Asimismo, el sistema debe restringir cada grado de libertad a girar según los ángulos mostrados en el Cuadro 3.1, con respecto al plano horizontal y mediante la respuesta ante activación de sensores de final de carrera asignados, acoplados a las articulaciones.

Cuadro 3.1: Ángulos de giro permitidos

Grado	Ángulo mínimo	Ángulo máximo
M1	$\pm 360^\circ$	
M2	-45°	225°
M3	-135°	135°
M4	-135°	135°
M5	$\pm 360^\circ$	

El diseño del software del sistema completo, realizado en la presente tesis, se enfoca en lograr la comunicación entre las distintas etapas de hardware del sistema electrónico (maestro, esclavos y computador), la programación de los microcontroladores para realizar el movimiento y la implementación de la interfaz de usuario. El diseño del hardware forma parte de otra tesis denominada “Diseño e implementación del hardware del sistema electrónico de un robot manipulador de cinco

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

grados de libertad”; en ella, se diseña, se elige y se implementa el hardware requerido para el funcionamiento del sistema integral; lo cual comprende: elección de sensores, elección de los excitadores, implementación del bloque de potencia, elección de la fuente de de alimentación, elección de los microcontroladores y diseño e implementación de la tarjetas para el modelo maestro-esclavos elegido.

3.2. Diagrama de bloques

En la Figura 3.1 se presenta el diagrama de bloques que se utilizó para el diseño y desarrollo del sistema integral implementado, el cual es constituido por la Planta, Hardware y Software. Posteriormente se detallan las características del hardware heredado, así como también los requerimientos y el desarrollo del software de los bloques Interfaz de Usuario, Control Maestro y Control Esclavo.



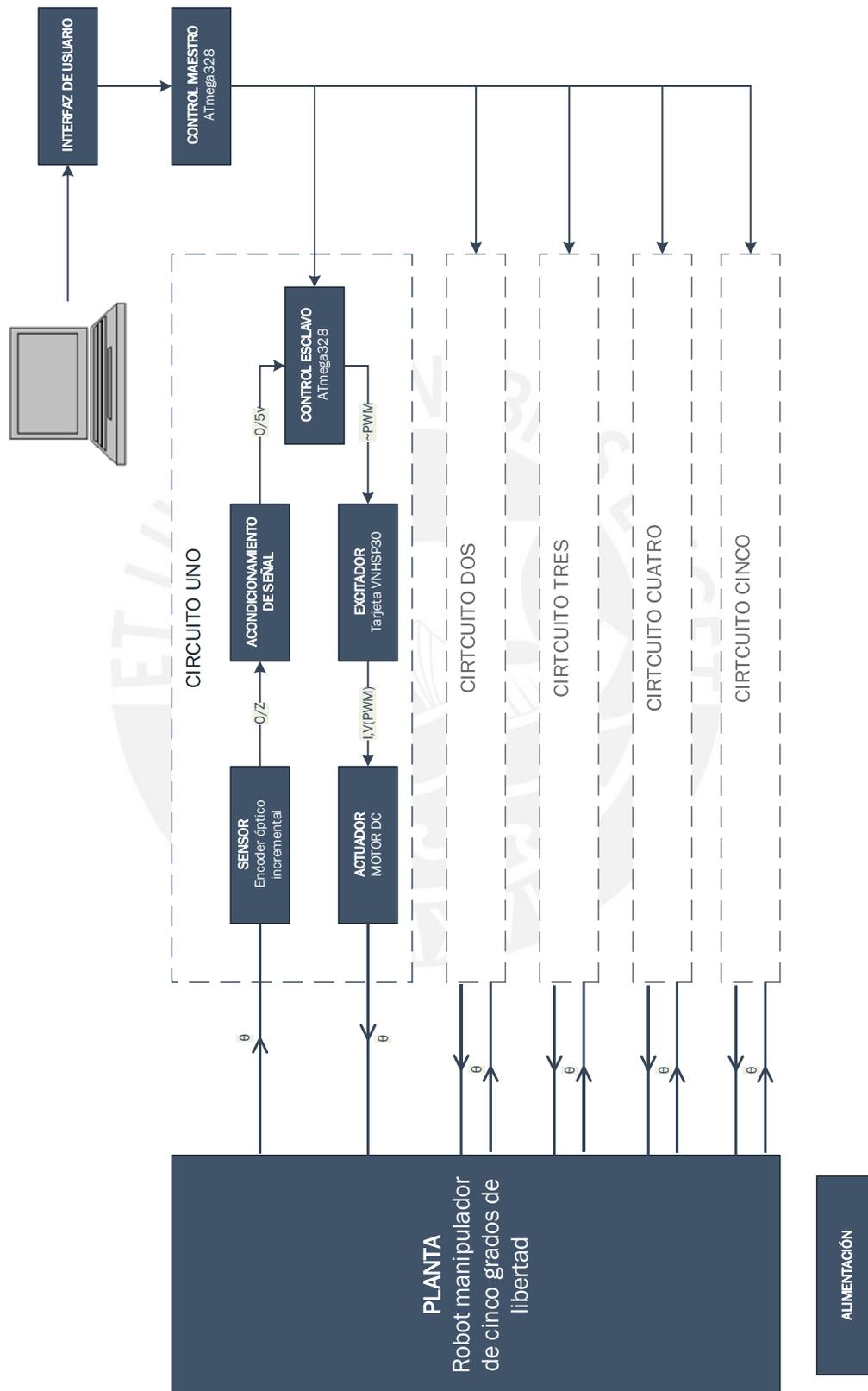


Figura 3.1: Diagrama de bloques

3.3. Hardware

En este acápite se exponen las características del hardware que ha sido heredado para el presente trabajo de tesis.

3.3.1. Planta

La planta sobre la cual se va a trabajar es una estructura metálica de 27.41Kg que contiene a los sensores y actuadores. El sistema está diseñado para soportar una carga máxima de 1.0Kg. En el Cuadro 3.2 se indica las distancias entre los ejes de giro de los grados de libertad del robot (Ver Anexo 2). La Figura 3.2 muestra las dos primeras articulaciones del robot manipulador.

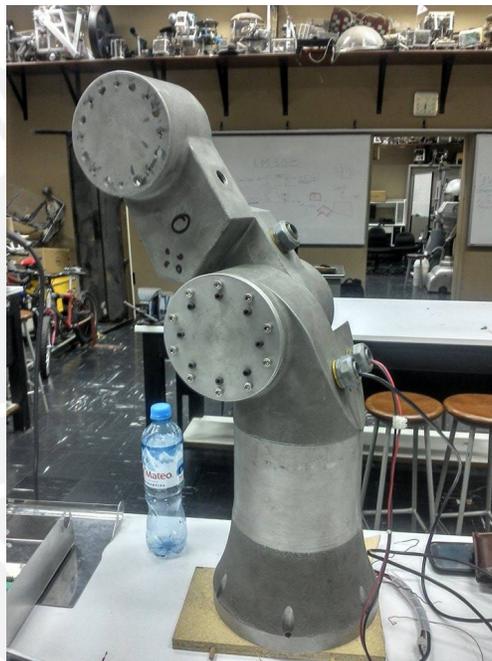


Figura 3.2: Robot manipulador - Articulación 1 y 2 ensambladas

Cuadro 3.2: Distancia entre los orígenes de los ejes de los motores

Eje	Distancia (m)
L1	0.175
L2	0.280
L3	0.280
L4	0.190
L5	0.090

Fuente: Elaboración propia

3.3.2. Actuadores

Tal como se mencionó en la Sección 3.1, los actuadores del sistema para cada grado de libertad han sido heredados. Estos son motores de corriente continua cuyas características se describen en el Cuadro 3.3 (Ver anexos 3, 4, 5 y 6).

Cuadro 3.3: Características de los actuadores. Fuente: Hojas de datos

Actuador	Modelo	Marca	Voltaje recomendado (V)	Máx. RPM recomendado
M1	S23K100E	Magmotor	12-120	4000
M2	S28K100E	Magmotor	12-120	4000
M3	S28K100E	Magmotor	12-120	4000
M4	S15O150E	Magmotor	12-120	4000
M5	8693	Pittman	10-48	4000

Fuente: Elaboración propia

Es necesario mencionar que para cada actuador existe un reductor mecánico del tipo *Harmonic Drive*, con un factor establecido en el Cuadro 3.4 (Ver Anexo 18). Dado que dichos factores de reducción son muy elevados, para efectos de control, se consideró que las articulaciones de la planta están desacopladas [20].

Cuadro 3.4: Harmonic Drives y las constantes de reducción.

Actuador	Código de Harmonic Drive	Factor de reducción
M1	FB-25-120-2-GR	120
M2	FB-32-160-2-GR	160
M3	FB-25-120-2-GR	120
M4	FB-20-130-2-GR	130
M5	FB-14-100-2-GR	100

Fuente: Elaboración propia

3.3.3. Excitadores

Para la manipulación de la corriente de los actuadores se dispone de tres excitadores **Dual VNH2SP30 Motor Driver Carrier MD03A** (Figura 3.3) [5]; dicha tarjeta está basada en el circuito integrado VNH2SP30. Este excitador, además de las características eléctricas propias de la tarjeta (Cuadro 3.3), posee las siguientes características adicionales de protección:

- Desactivación por alta temperatura
- Desactivación por sobre y bajo voltaje
- Desactivación ante corto-circuito
- Desactivación ante desconexión de V_{cc} o Gnd

- Enclavamiento para sobre voltajes

Adicionalmente, el excitador, posee un sensor de corriente que entrega un voltaje proporcional a la corriente consumida por el actuador en manipulación; dicho sensor se usa para desactivar las articulaciones correspondientes en caso de detección de sobrecargas.

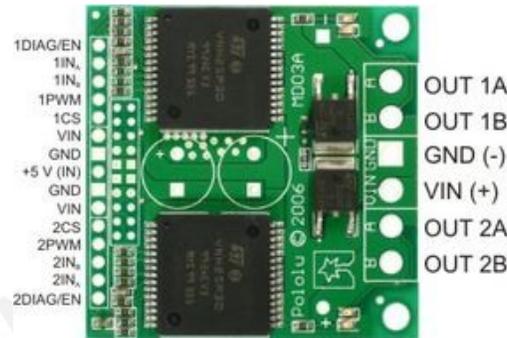


Figura 3.3: Tarjeta del excitador VNH2SP30 MD03A [5]

Cuadro 3.5: Características del excitador. Fuente: Elaboración propia

Item	Valor
Rango de voltaje (V)	5.5 - 16
Corriente admitida (A)	14
Corriente pico	30
Tamaño	1.86" × 1,86"
Canales	2

Para evitar trabajar en el límite de voltaje del excitador, se opera a 15 voltios como máximo en todo momento.

3.3.4. Sensores

En esta etapa se describen las características de los sensores utilizados en el presente trabajo de tesis, estos son: los sensores de posición angular utilizados (*encoders*), los sensores de final de carrera y los sensores de corriente.

3.3.4.1. Sensor de posición angular

Para el sensado de la posición angular de los actuadores se heredaron los *encoders* incrementales descritos en el Cuadro 3.6 (Ver Anexos 7, 8 y 9).

Cuadro 3.6: Características de los *encoders* recibidos

Item	Alternativas		
	M1,M2 y M3	M4	M5
Actuador	M1,M2 y M3	M4	M5
Modelo	M53-Dynapar	M15-Dynapar	HEDS-9140 Avago
Eje(axial)	Axial	Axial	Axial
Acoplamiento	3/8"	1/4"	0.15"
Voltaje(V)	5	5	5
PPR	2048	1024	500

Es preciso mencionar que todos los *encoders* utilizados poseen salidas del tipo *Open-Collector*, por lo que las tarjetas poseen las correspondientes resistencias de *pull-up* con un valor de 2K Ohm.

3.3.4.2. Sensor de máxima posición angular

Para detectar que se ha llegado a la máxima posición angular -físicamente posible- para los grados 2, 3 y 4 descritos en el Cuadro 3.1, se utilizaron sensores de final de carrera (Figura 3.4). Estos son pulsadores diseñados para cortocircuitarse ante el contacto entre superficies.



Figura 3.4: Sensor de final de carrera

3.3.4.3. Sensor de corriente

Como se mencionó en la Sección 3.3.3, el excitador posee un sensor de corriente que entrega de forma lineal 0.13 voltios por cada amperio de corriente que recibe el actuador [5]. En caso se detecte sobre corriente -producida por una sobre carga-, la articulación en cuestión dejará de moverse.

3.3.5. Controladores

En esta sección se describe el hardware utilizado sobre el cual se desarrolló el software para realizar el control del sistema.

La solución se dividió en dos partes: Circuito maestro y circuitos esclavos.

El circuito maestro recibe los datos de la PC e indica a los esclavos el movimiento a realizar. Los esclavos ejecutan directamente el giro a realizar manipulando el excitador asignado, según lo solicitado por el maestro. Para esto se utiliza un microcontrolador como unidad de procesamiento de cada circuito propuesto.

El microcontrolador sobre el cual se va a desarrollar el software del esclavo y del maestro es un *ATmega8A* de la marca *Atmel* (Ver Anexo 10). Este cumple las características mínimas necesarias (Ver Cuadro 3.7) para el correcto funcionamiento del sistema.

Cuadro 3.7: Características del microcontrolador *ATmega8A*

Característica	ATmega8A
Pines E/S	24
Clock	0-20Mhz
FLASH/EEPROM/SRAM	32K/1K/2K
Comunicación	TWI, UART
ADC	8 Canal
Interrupciones externas	2

Fuente: Elaboración propia

3.3.5.1. Circuito maestro

La interfaz de comunicación entre la PC y el circuito maestro se implementó mediante el protocolo serial *UART*, para esto se utilizó un módulo conversor de USB a Serial TTL (Figura 3.5).



Figura 3.5: Conversor USB a Serial TTL

En la Figura 3.6 se muestra el diagrama esquemático correspondiente al circuito maestro, el cual es detallado a continuación. Para el procesamiento se eligió el procesador *ATmega 8A* de la familia *AVR* de la marca *Atmel*. Para la comunicación con los esclavos se utiliza el protocolo *I²C*, dicho protocolo permite manipular múltiples esclavos según direcciones previamente asignadas. Las líneas de este protocolo son *SDA*, *SCL* y *GND*. En la tarjeta se encuentran también los pines del protocolo *SPI*, pertinentes para grabar el microcontrolador de forma sencilla.

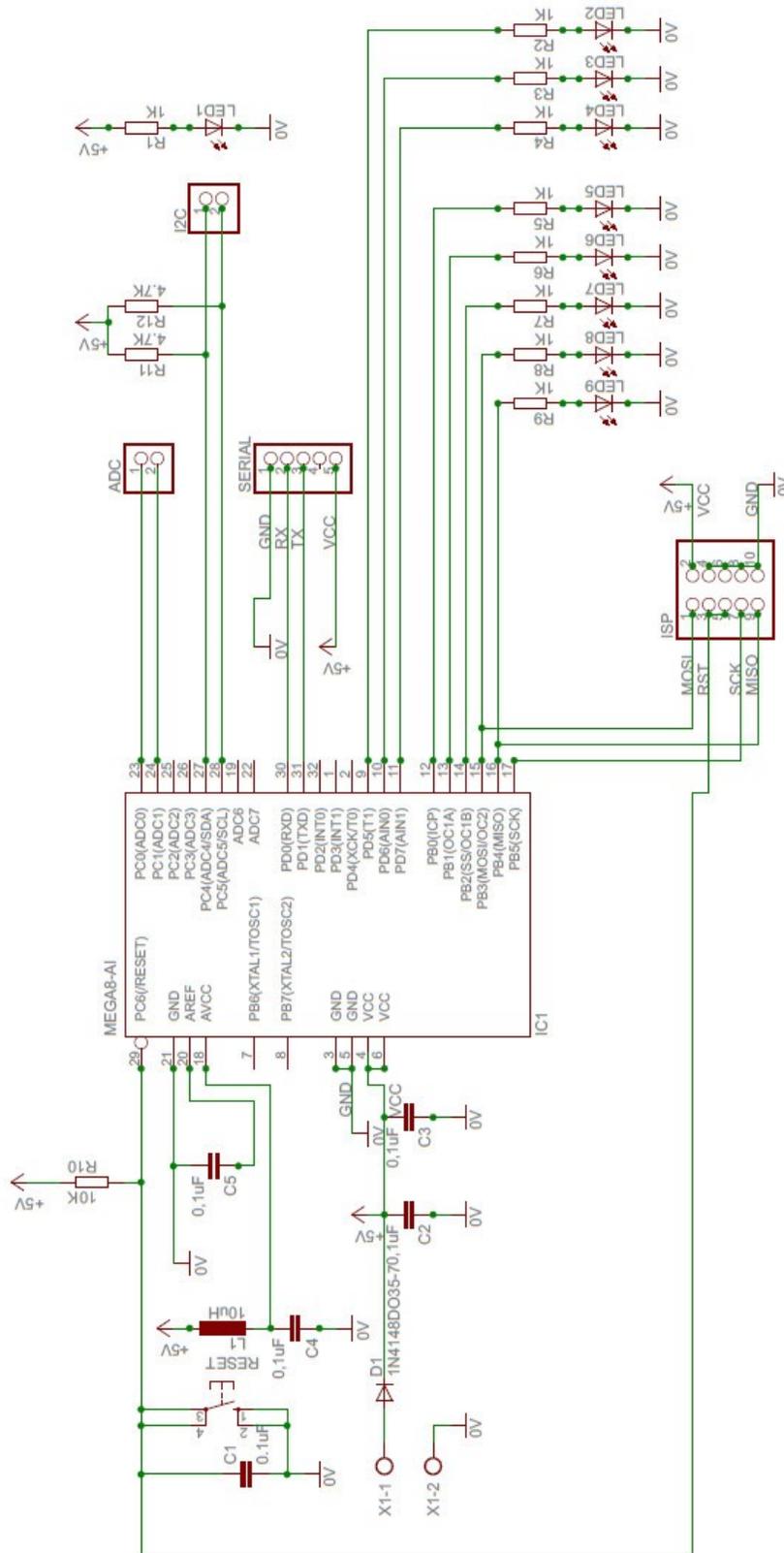


Figura 3.6: Diagrama esquemático de la tarjeta Maestro

3.3.5.2. Circuitos esclavos

El presente circuito (Figura 3.7) fue replicado cinco veces, pues cada tarjeta es capaz de controlar solo un actuador. De igual forma que en el circuito maestro, se eligió como unidad de procesamiento al *ATmega 8A*. La tarjeta recibe las señales entregadas por los *encoders* y tiene como salidas los cuatro pines requeridos para manipular el excitador, los cuales son: *EN*, *INA*, *INB* y *PWM*. Además, al igual que la tarjeta maestra, posee los pines necesarios para los protocolos *I²C* para la comunicación con el maestro y *SPI* para facilitar el grabado del software en los microcontroladores.



3.4. Software

En esta sección se detalla el desarrollo del software que define la lógica de funcionamiento de cada etapa específica del sistema; asimismo, también se describe el desarrollo del algoritmo de control implementado en los microcontroladores esclavos. El sistema debe recepcionar los ángulos ingresados por el usuario y, seguidamente, ejecutar la acción de control en los actuadores solicitados a partir de los parámetros del algoritmo de control y la retroalimentación recibida por los *encoders*, para así lograr que el error sea el mínimo posible.

El Software ha sido dividido en tres partes: Software de la interfaz de usuario, software del microcontrolador maestro y software de los microcontroladores esclavos.

La programación de los microcontroladores fue realizada en lenguaje C a través del entorno de programación *Atmel Studio 6* y la interfaz de usuario se desarrolló en el lenguaje C#, haciendo uso del entorno de programación *Visual Studio 2013* de *Microsoft*.

3.4.1. Interfaz de usuario

La interfaz de usuario debe poseer la siguientes características:

- Permitir la elección del puerto para la comunicación serial y aperturar la conexión de la misma.
- Recepcionar los ángulos que el usuario requiere que gire cada actuador seleccionado del robot manipulador; estos deben ser enviados vía serial según la necesidad del usuario.
- Mostrar la posición en coordenadas cartesianas del actuador final, a partir de los ángulos ingresado por el usuario (cinemática directa).

Para dicho fin se resolvió utilizar el lenguaje C#; este es un lenguaje de la familia .NET de Microsoft que permite la creación de interfaces bajo el entorno de programación *Visual Studio*, el cual otorga una serie de herramientas que facilitan de forma considerable el desarrollo de las mismas [9]. Por otro lado, C# es un lenguaje derivado de C lo cual reduce la curva de aprendizaje requerida para el desarrollo de la presente interfaz. (Ver Anexo 11)

3.4.1.1. Comunicación serial

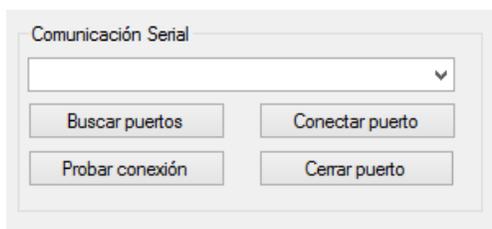


Figura 3.8: Etapa de comunicación serial de la interfaz

En la Figura 3.9 se muestra el diagrama de flujo que describe la lógica implementada para los métodos asignados a los eventos. Cada evento se genera al presionar cada uno de los cuatro botones creados para la etapa de comunicación serial de la interfaz de usuario (Figura 3.8).

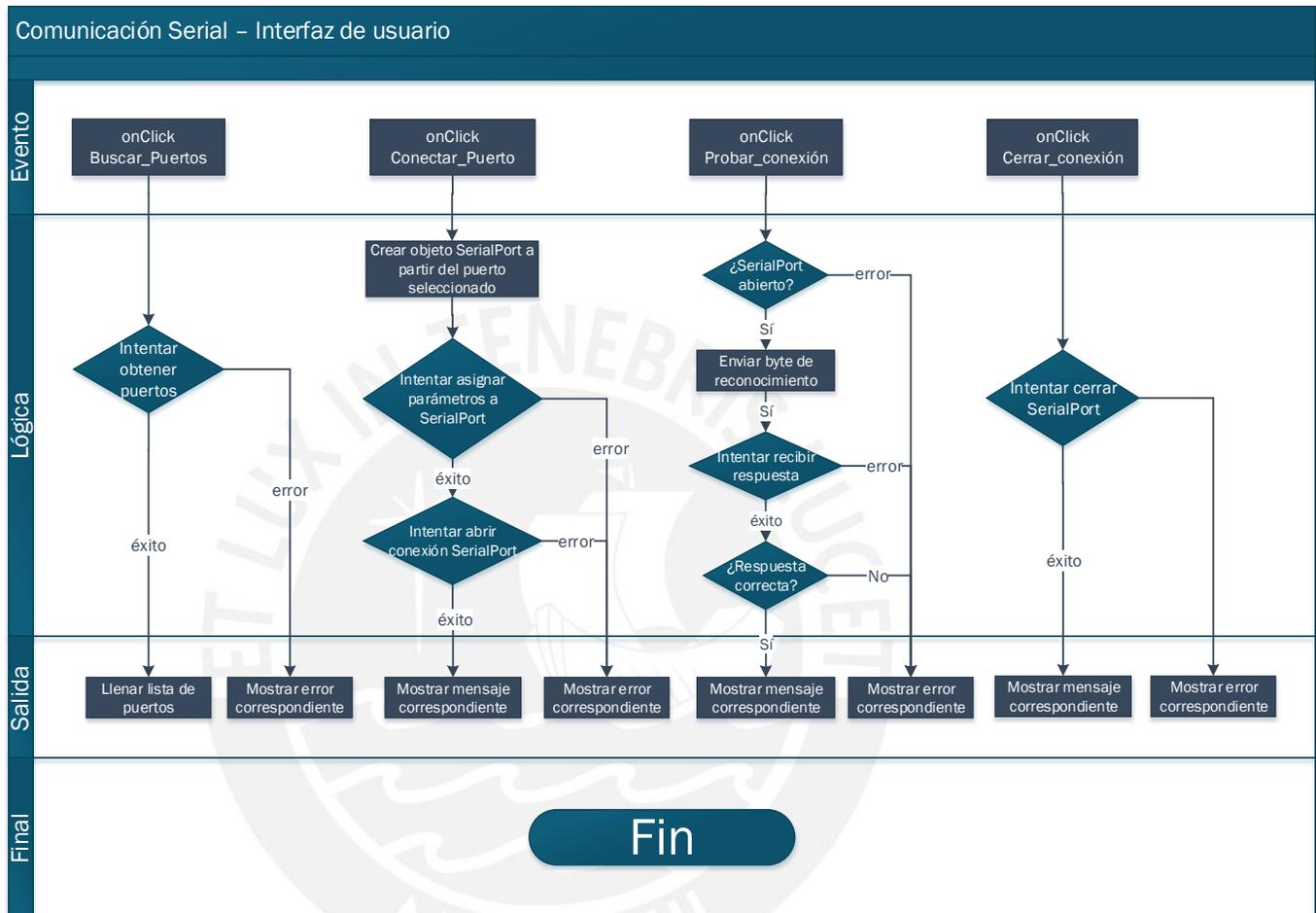


Figura 3.9: Diagrama de flujo de los eventos asociados a la Comunicación Serial

Las condicionales de intento fueron implementadas mediante las funciones -propias del lenguaje C#- *try* y *catch*, con el fin de controlar posibles excepciones del programa y mostrar mensajes de error o de éxito.

La configuración de módulo serial se realiza a una velocidad de 9600 bits por segundo pues como máximo solo es necesario enviar 27 bytes para mover las cinco articulaciones (Ver Cuadro 3.8), por lo que la velocidad elegida es aceptable para la aplicación. Adicionalmente se configuraron los siguientes parámetros de control y detección de errores:

- Velocidad: 9600bps
- Paridad: Par
- Bits de parada: 1
- Bits de datos: 8

- Timeout de escritura: 500
- Timeout de lectura: 500

3.4.1.2. Envío de trama

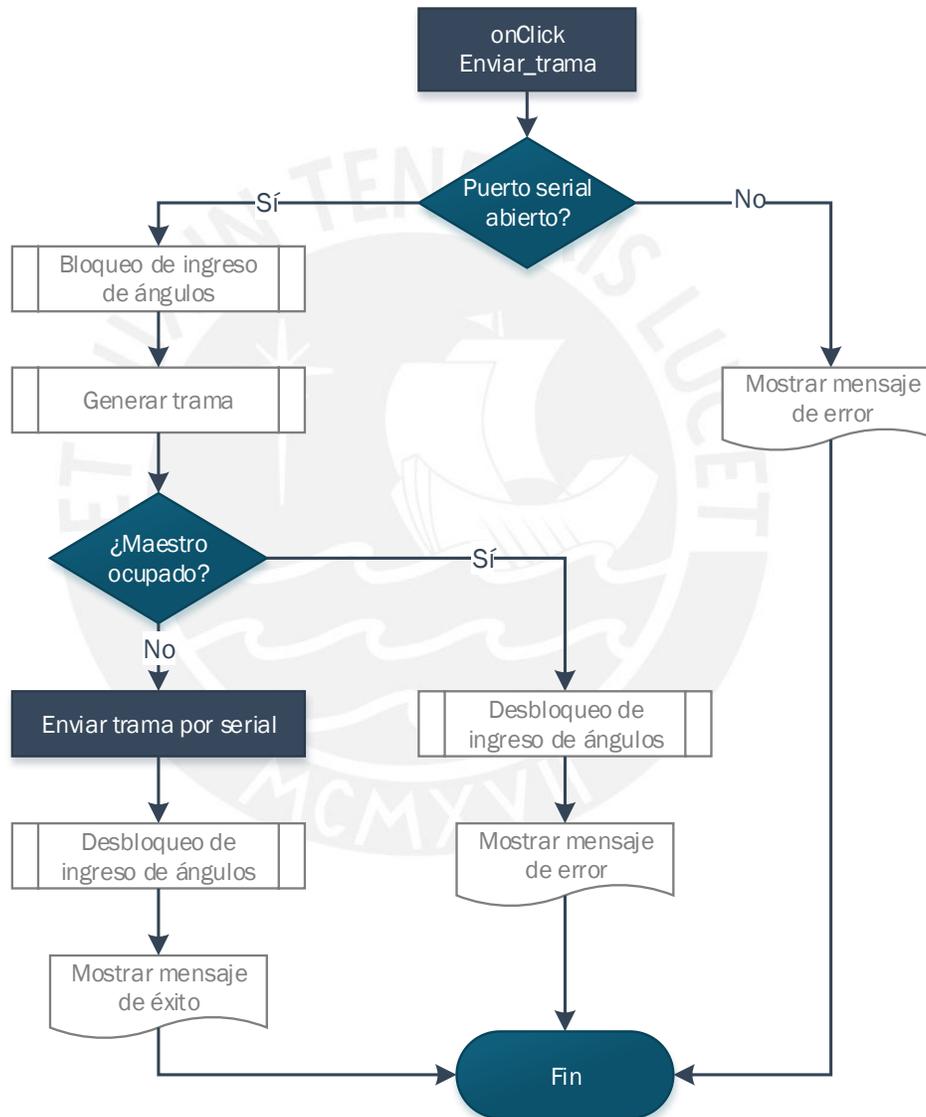


Figura 3.10: Diagrama de flujo del envío de trama vía serial

La trama es enviada solo cuando el microcontrolador maestro no se encuentra ocupado; para esto, se creó una variable que almacena dicho estado, la cual es actualizada cuando la interfaz envía un byte de reconocimiento de estado al maestro, este responde indicando si está libre u ocupado para recibir.

Para generar la trama se leen, en una estructura iterativa, los valores del arreglo de entradas numéricas el cual contiene los valores de las posiciones angulares de las articulaciones ingresadas por el usuario. Si el ángulo que se está leyendo se encuentra con su respectiva casilla seleccionada se adiciona su valor a la trama; caso contrario, se ignora. El ángulo se multiplica por 10 para poder enviar los valores decimales de forma simple; se envía a cada esclavo bajo la estructura representada en el Cuadro 3.8.

Cuadro 3.8: Ejemplo de trama enviada por serial

Inicio	Articulación	Ángulo				...	Articulación	Ángulo				Fin
\$	A	1	8	0	5	...	E	0	9	0	7	@

3.4.1.3. Cinemática directa

Para calcular la posición a partir de la cinemática directa del sistema, se multiplicaron las matrices de transformación homogénea de articulación a articulación para así llegar a la ubicación final de la carga del robot manipulador; para esto se utilizó la representación de Denavit-Hartenberg (DH) cuyos parámetros están descritos en el Cuadro 3.9 (Ver Anexo 1).

Cuadro 3.9: Parámetros de Denavit-Hartenberg

Articulación	α	a	θ	D
1	$\pi/2$	0	θ_1	$L1$
2	0	$L2$	θ_2	0
3	0	$L3$	θ_3	0
4	$-\pi/2$	θ_4	0	0
5	0	0	θ_5	$L4+L5$

Fuente: Elaboración propia

Cada cambio en las entradas numérica genera un evento al cual se le asignó un método que recalcula la posición de la carga final del robot. Dicho método lee el arreglo que contiene los parámetros DH de cada articulación y, a partir de estos, calcula cada una de las matrices de transformación; finalmente, se multiplican las cinco matrices obtenidas y, de la matriz generada, se leen las coordenadas cartesianas de la última articulación, las cuales son mostradas en tres cuadros numéricos (Figura 3.11).

Cinemática Directa

Coordenada X

Coordenada Y

Coordenada Z

Figura 3.11: Etapa de cinemática directa

3.4.2. Software del microcontrolador Maestro

El microcontrolador maestro debe ser capaz de interpretar la trama recibida vía serial de la interfaz y, según esta, enviar los datos correspondientes a cada uno de los esclavos.

En la Figura 3.12 se muestra el diagrama de flujo principal implementado en el microcontrolador maestro. (Ver Anexo 12)

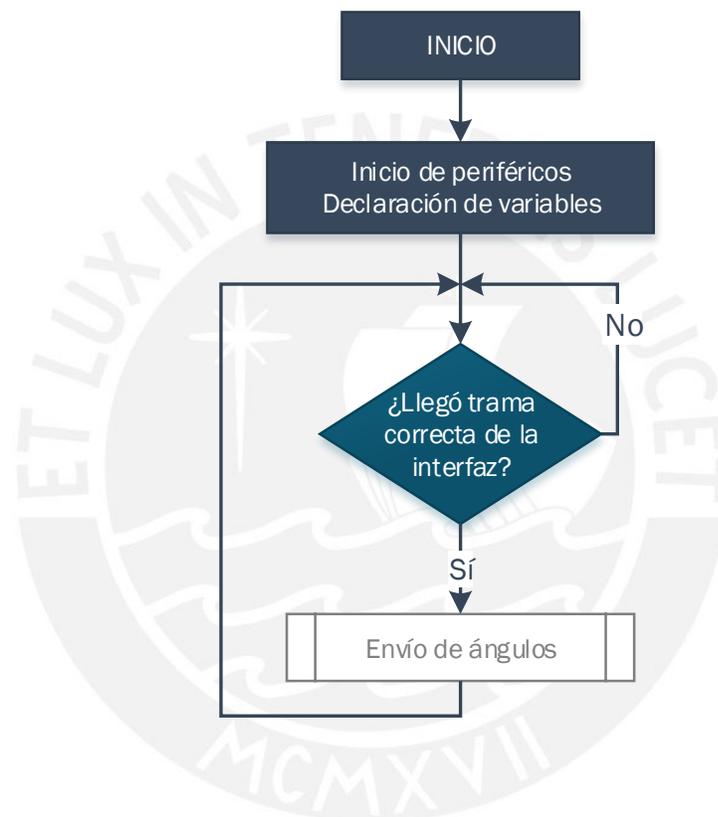


Figura 3.12: Diagrama de flujo principal del microcontrolador Maestro

Inicialmente se configuran los siguientes módulos:

- Los pines de salida, asignados en el diagrama esquemático (Figura 3.6). Es preciso mencionar que no se tienen pines de entradas de propósito general; no obstante, sí se utilizan pines de entrada correspondientes a los periféricos del microcontrolador.
- El módulo I^2C del microcontrolador en modo maestro a la velocidad de reloj estándar de 100Khz.
- Módulo $UART$ a 9600bps, 8 bits, paridad par y un bit de parada; configurado para interrupción por recepción para la comunicación con la PC. Se creó un *buffer* de 30 datos para la manipulación serial.

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

- Se declaran las variables a utilizar en el sistema; estas son las variables usadas por el bucle de espera que valida y recibe los datos recibidos por serial.

Dado que el microcontrolador maestro recibe de la Interfaz de usuario los ángulos que este requiere que cada articulación gire, el programa se queda en un bucle comprobando que haya llegado la trama correcta (Ver Cuadro 3.8).

Una vez recibidos los datos, se sale del bucle y se envían a los esclavos correspondientes bajo una trama similar a la de la recepción vía serial: a partir de la dirección 0x00 del buffer I^2C que cada esclavo posee se escribe el ángulo solicitado en cuatro bytes y un final de comunicación @.

El envío de los ángulos a los esclavos se hace mediante la detección de estado ocupado; esta se realiza leyendo un byte del buffer I^2C -asignado para este propósito- del esclavo al que se está intentando enviar el ángulo de giro; si este no está disponible se envía la información correspondiente al siguiente esclavo y se llena un buffer del maestro en el que se encuentra el estado de los cinco grados de libertad. Luego de finalizado el intento de movimiento de los cinco grados, se lee de forma repetitiva -cada 100ms- el estado en el que se encuentran los grados que estaban ocupados para moverlos luego de que estos estén disponibles para recibir.

Luego de finalizado el envío de la trama a los esclavos, el programa regresa al bucle de comprobación de nuevos datos.

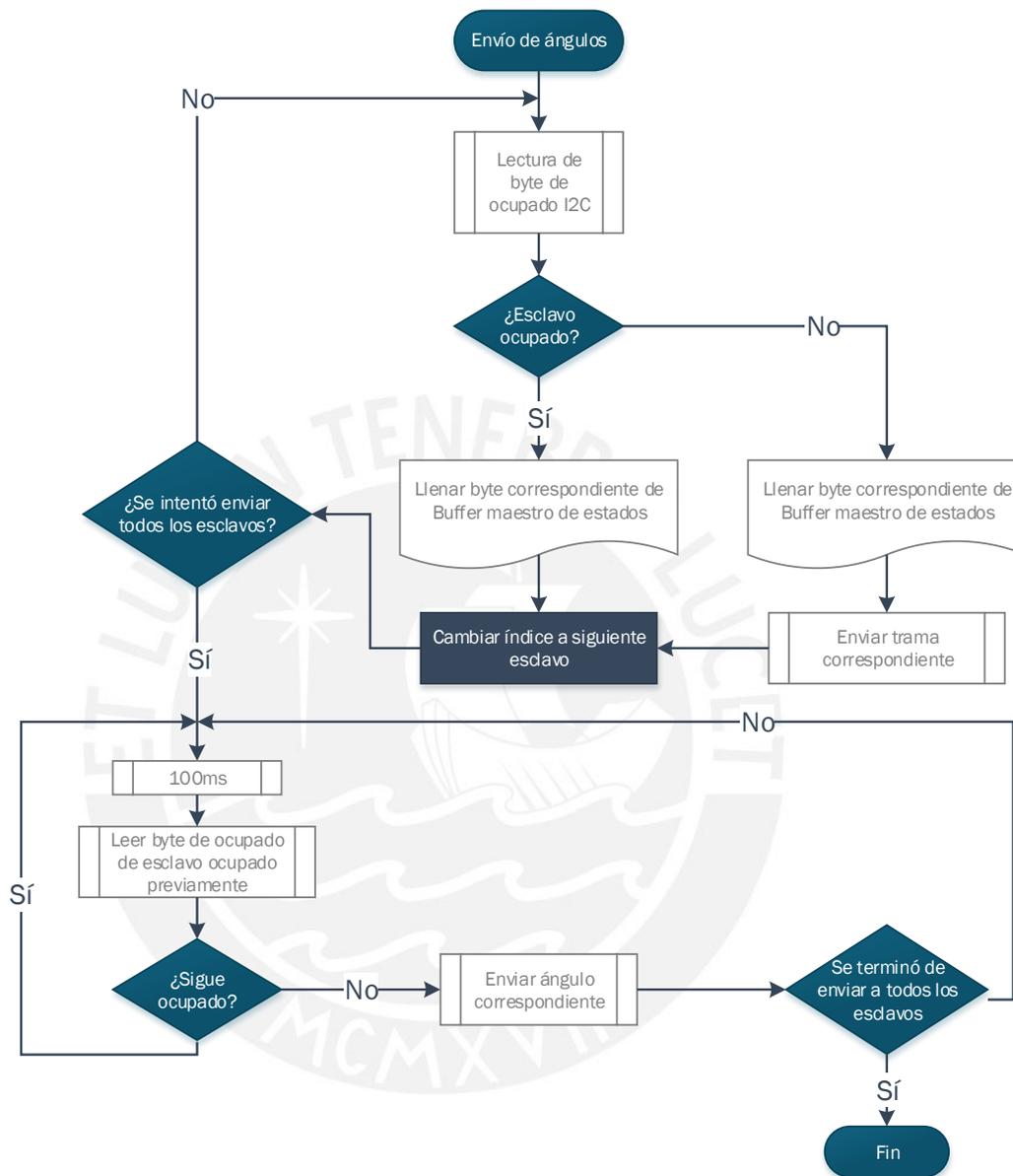


Figura 3.13: Diagrama de flujo del envío de ángulos a los controladores esclavos

3.4.3. Software de los microcontroladores Esclavos

Cada grado de libertad posee un circuito de control esclavo, el cual debe ser capaz de recibir del maestro la posición angular solicitada por el usuario y llegar a ella a partir del algoritmo de control retroalimentado. Asimismo, debe enviar -a petición del maestro- el estado en el que este se encuentra.

En la Figura 3.14 se muestra el diagrama de flujo principal implementado en los microcontro-

ladores esclavos. (Ver Anexo 13)

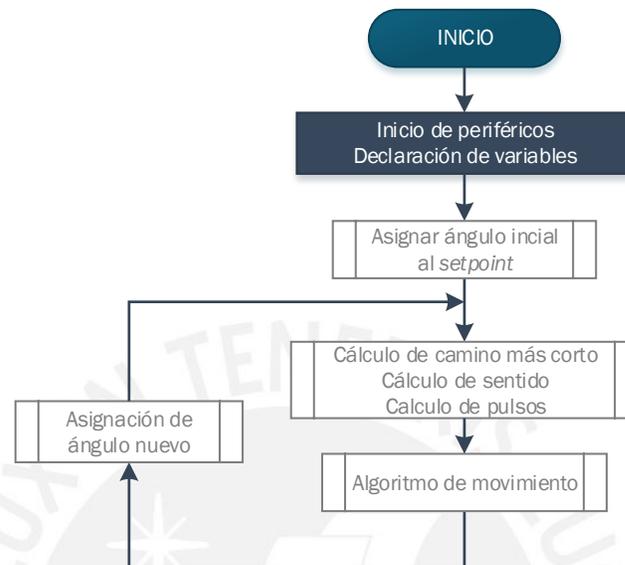


Figura 3.14: Diagrama de flujo principal de los microcontroladores Esclavo

Se inician los microcontroladores configurando los periféricos de la siguiente forma:

- Se configura el Timer 1 para la generación de la onda *PWM*. Se utilizó el modo 5 correspondiente a *FastPWM* de 8 bits con el prescalador en 8, lo cual da una frecuencia de onda de 3.906KHz en el pin *OC1A*. El ciclo de trabajo de dicha onda es manipulado por el controlador PID implementado, de esta forma se modifica el voltaje promedio que ingresa al actuador mediante un escalamiento previo desarrollado en la Sección 3.4.3.3; no obstante, este se saturará a la máxima alimentación con la que trabaja el excitador, que es 15 voltios.
- Se configura el módulo *I²C* para funcionar por interrupción como esclavo; es decir, este solo puede responder a la comunicación, mas no iniciarla. Se asigna un buffer de longitud 16 para la lectura o escritura del microcontrolador maestro.
- Se configura la interrupción externa *INT0* para realizar el seguimiento al *encoder* por flancos de subida.
- Se configura la interrupción externa *INT1* para la detección de la activación de los sensores de final de carrera en las articulaciones que así lo requieran.
- Se configura el módulo de comparación analógica para detección de sobrecorrientes teniendo como entrada el sensor de corriente que posee cada excitador.

- Se configura los pines de entrada y salida según el diagrama esquemático correspondiente (Figura 3.7)

Luego de configurados los periféricos del microcontrolador, se asigna el ángulo al cual se va a ir inicialmente (Cuadro 3.10) y, con este valor, se ingresa por primera vez al bucle principal del microcontrolador esclavo que se encarga de determinar el valor de la variable de control a partir del ángulo al que se quiere llegar (punto de referencia), el sentido de giro y el algoritmo de movimiento para realizar el giro de las articulaciones. En cada iteración del bucle principal se lee el arreglo que contiene al ángulo de referencia -pudiendo este ser un ángulo nuevo o no- y se lleva dicho valor a los algoritmos correspondientes, de forma tal que se realice el control del sistema durante todo el periodo de operación del robot, pues aún llegando al ángulo deseado podrían existir perturbaciones externas.

Cuadro 3.10: Posición inicial de los actuadores

Actuador	Ángulo
M1	0°
M2	45°
M3	-135°
M4	-45°
M5	0°

Fuente: Elaboración propia

Antes de iniciar los cálculos y el control del actuador se modifica un byte del buffer I^2C , que indica que el microcontrolador no ha llegado a la posición solicitada y se procede a realizar el movimiento del sistema. Se obtiene el punto de referencia (SP_n) mediante la Ecuación 3.1, a partir de la cantidad de pulsos por revolución (PPR_n) del *encoder* y la diferencia angular ($\Delta\theta$) de la posición solicitada y la posición anterior.

$$SP_n = \frac{\Delta\theta \times PPR_n}{360} \quad (3.1)$$

En el caso de M1 y M5 se obtiene el sentido y el punto de referencia teniendo en cuenta el camino que otorgue la menor cantidad de pulsos para llegar al ángulo solicitado. Para el resto de articulaciones solo existe un único sentido posible para llegar a cualquier posición, pues estos se encuentran físicamente limitados en su rotación (Ver Cuadro 3.1). A partir de estos datos calculados se inicia el algoritmo de movimiento (Figura 3.15).

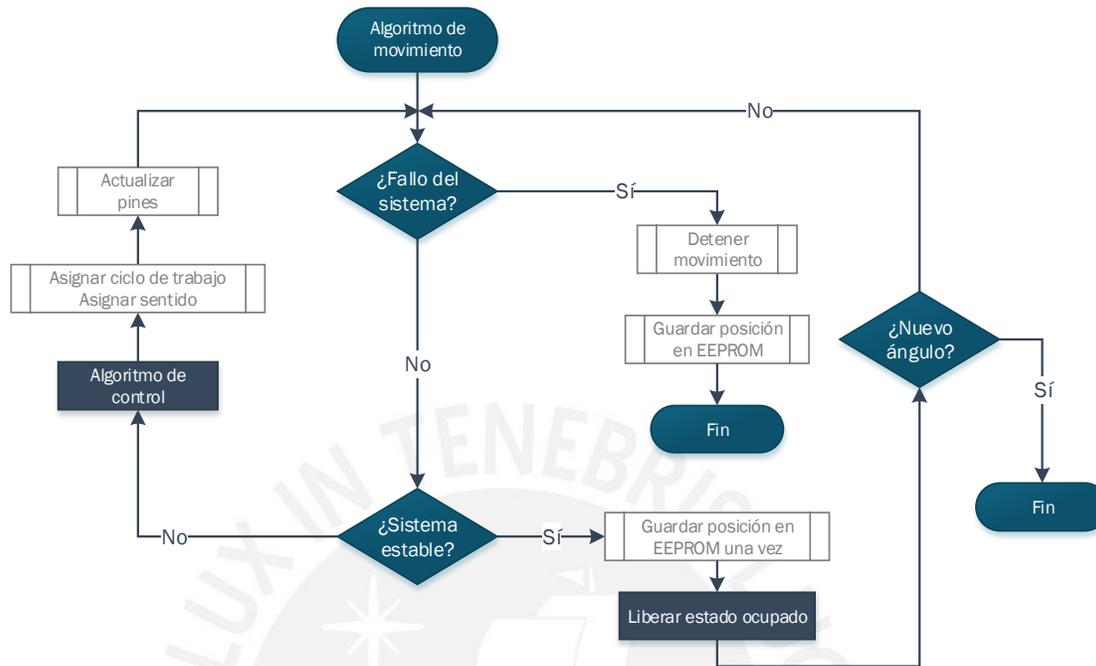


Figura 3.15: Diagrama de la subrutina encargada del algoritmo de movimiento

Al inicio se ingresa al bucle principal de la subrutina y se procede a leer la variable de error que es modificada por las subrutinas de interrupción que detectan sobrecorriente (comparación analógica) o por final de carrera (interrupción externa); en caso de error se detiene el movimiento, se guarda la posición hasta la que se llegó en la *EEPROM* y se finaliza la subrutina; en caso de no presentarse error y de que el sistema aún no esté estable, se efectúa la subrutina encargada de realizar el algoritmo de control de la planta, se asigna el valor al ciclo de trabajo de la onda *PWM* que manipula el voltaje promedio del actuador y se actualizan los pines que manipulan el sentido de giro del actuador. Se considera que el sistema es estable cuando el error es cercano a cero durante un tiempo de 500 veces el periodo de muestreo, esto es determinado en la sección Sistema Estable, encontrada líneas adelante.

Una vez que el sistema ha alcanzado la estabilidad se permite nuevamente recibir por *I²C* la trama que contenga un nuevo ángulo, modificando el byte asignado a estado libre. Si se recibe un nuevo ángulo se sale de la subrutina y se regresa al bucle principal; caso contrario, se continúa controlando el sistema ante posibles perturbaciones. Asimismo, en estabilidad se activa también la interrupción por comparación analógica para detectar posibles fallos por sobrecorriente.

3.4.3.1. Algoritmo de control

El algoritmo de control utilizado debe garantizar que cada actuador llegue a la posición requerida por el usuario, reduciendo el error al mínimo posible. Para esto, se diseñaron e implementaron

cinco controladores monoarticulares PID, bajo el diagrama de bloques mostrado en la Figura 3.16.

Para el presente trabajo de tesis se consideró que los parámetros variables del modelo dinámico completo del sistema -o torque de perturbación (τ_p)- que son: influencia de la gravedad, elementos no constantes de la matriz de inercias y fuerza centrípeta, no afectan el modelo del sistema pues cada una de las articulaciones posee un elevado factor de reducción [20] (Ver Cuadro 3.4).

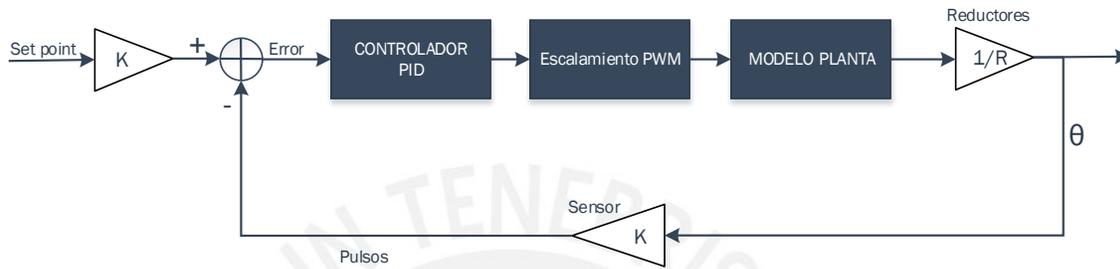


Figura 3.16: Diagrama de bloques de control

Para tal propósito, se modeló el sistema y se diseñó el controlador PID utilizando el *toolbox Simulink* de *Matlab*. Los parámetros del controlador se obtuvieron mediante ajustes manuales realizados al controlador simulado.

Modelamiento del sistema:

El modelo se obtiene a partir del circuito equivalente del actuador, representado en la Figura 3.17.

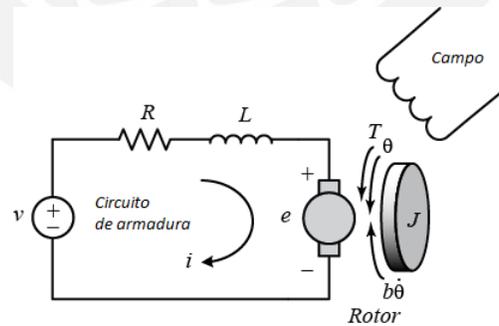


Figura 3.17: Diagrama del modelo de un motor DC [4]

Realizando el análisis eléctrico de la malla correspondiente al modelo (Figura 3.17) se obtiene la Ecuación 3.2.

$$La \times \frac{di}{dt} + Ra * i + Ke \times \dot{\theta} = v(s) \tag{3.2}$$

Donde La es la inductancia del bobinado; Ra , la resistencia del bobinado; i , la corriente de

armadura; K_e , la constante de velocidad; K_t , la constante de torque; y $\dot{\theta}$, la velocidad angular del rotor.

Dadas las consideraciones asumidas, las ecuaciones de momento de torque de cada articulación se reducen al modelo de los actuadores afectados solo por el momento producido por la inercia (J_{planta}), el rozamiento viscoso (B) y el torque de perturbación (τ_p) dada la estructura del robot, siendo estos referenciados a los ejes de los actuadores a través de los reductores que estos poseen. Luego, por la segunda ley de Newton se obtiene la Ecuación 3.3.

$$J_{planta} \times \ddot{\theta} + B \times \dot{\theta} + \tau_p = Kt \times i \tag{3.3}$$

Reemplazando la Ecuación 3.2 en la Ecuación 3.3, eliminando el torque de perturbación y transformándolo al dominio de Laplace, se tiene que la ecuación de transferencia para cada articulación del sistema es:

$$\frac{\theta(s)}{v(s)} = \frac{Kt}{[(La * s + Ra) (Ja * s + B) + Kt * Ke] * s} \tag{3.4}$$

Cálculo de coeficientes

Para obtener los modelos de los actuadores es necesario conocer los coeficientes propios de cada articulación, estos fueron hallados de la siguiente manera: Las resistencias e inductancias de armadura y los coeficientes de rozamiento viscoso se encontraron mediante ensayos realizados; las constantes de torque-velocidad fueron tomadas de las hojas de datos de los actuadores; y, finalmente, los momentos de inercia se calcularon a partir de las masas manipuladas por cada actuador y las distancias de estos a los centros de masa.

Resistencias e inductancias de los actuadores

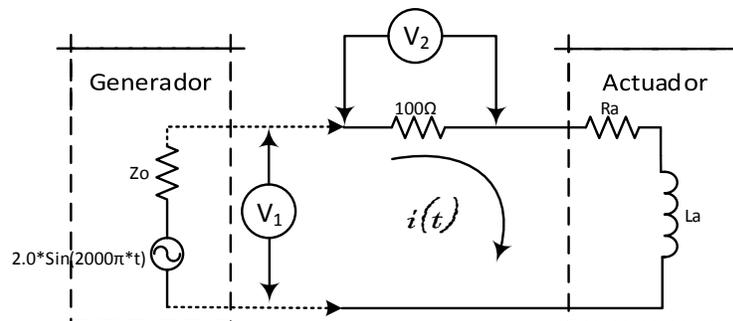


Figura 3.18: Circuito utilizado para la medición de parámetros

Para hallar las resistencias (R_a) e inductancias de armadura (L_a) se realizó un ensayo para cada actuador el cual consistió en conectar una fuente de voltaje alterno en serie con una resistencia

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

de prueba (R_{prueba}) y con el actuador (Figura 3.18) para así obtener, de forma indirecta, el valor de la corriente que circula por el circuito en todo instante. El valor de la amplitud de la corriente está dado por el voltaje medido en la resistencia agregada y dividido entre su valor; por otro lado, el desfase se obtiene a partir de la lectura de la diferencia de fases de la señal de entrada y la señal medida en la resistencia de prueba, visualizadas en el osciloscopio (Figura 3.19). Del circuito se obtienen las ecuaciones 3.5 y 3.6, donde v_1 es la magnitud del voltaje de entrada; v_2 , la magnitud del voltaje medido en R_{prueba} ; ϕ , el desfase de la corriente medida con respecto al voltaje entregado y ω , la frecuencia angular de la señal de entrada.

$$\frac{v_1}{v_2/R_{prueba}} = \sqrt{(\omega La)^2 + (R_{prueba} + Ra)^2} \tag{3.5}$$

$$\phi = \tan^{-1} \left(\frac{X_L}{R_{prueba} + Ra} \right) \tag{3.6}$$

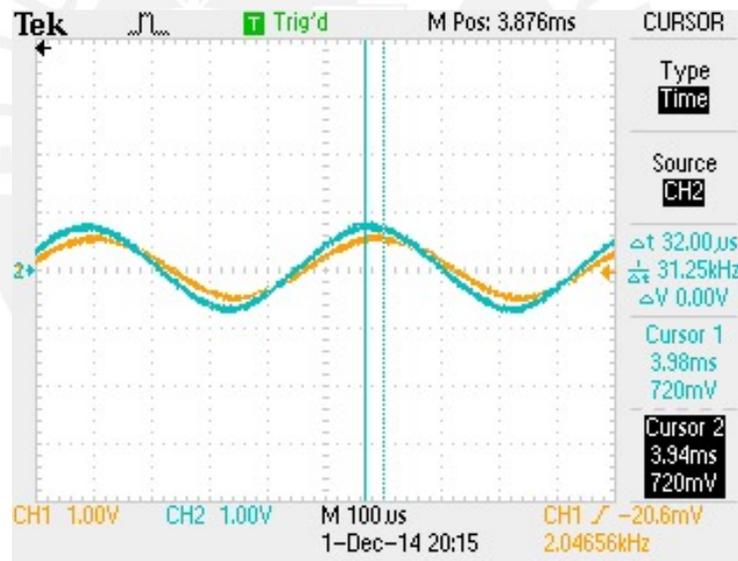


Figura 3.19: Desfase de ondas en el ensayo para el actuador M1

Las condiciones de prueba fueron: $\omega = 4000\pi$ rad/s, $R_{prueba} = 100\Omega$ y la impedancia del generador de 50Ω . Se aseguró que el voltaje de los actuadores no se encuentren en movimiento para así evitar sobrecorrientes.

A partir de las ecuaciones mencionadas se obtuvieron los valores de Ra y La para todos los actuadores, los cuales se muestran en el Cuadro 3.11.

Cuadro 3.11: Cálculo de resistencias e inductancias de armadura

Actuador	v_2/R_{prueba}	Ra (Ohm)	La (mH)
M1	$0,0116 * \text{sen}(4000\pi t + 21,70^\circ)$	24.95	3.956
M2	$0,0112 * \text{sen}(4000\pi t + 31,49^\circ)$	20.87	5.287
M3	$0,0111 * \text{sen}(4000\pi t + 31,51^\circ)$	20.90	5.252
M4	$0,0130 * \text{sen}(4000\pi t + 43,86^\circ)$	21.93	0.174
M5	$0,0145 * \text{sen}(4000\pi t + 18,08^\circ)$	4.12	2.640

Fuente: Elaboración propia

En el caso particular de M5 la hoja de datos indica el valor de resistencia e inductancia de armadura siendo estos $4,02\Omega$ y $2,440mH$, respectivamente. Estos valores son muy cercanos a los hallados mediante el ensayo realizado.

Constantes de torque-velocidad de los actuadores

En el Cuadro 3.12 se muestran las constantes inherentes a los actuadores tomadas de las hojas de datos respectivas. Es necesario resaltar en la constante de velocidad (Ke) y la constante de torque (Kt) son numéricamente iguales para todos los actuadores.

Cuadro 3.12: Constantes de torque y voltaje

Actuador	$Kt = Ke$
M1	0.141
M2	0.257
M3	0.257
M4	0.031
M5	0.024

Fuente: Elaboración propia

Rozamiento viscoso

El cálculo del coeficiente de rozamiento viscoso del sistema actuador-reductor se realizó mediante un ensayo derivado de la Ecuación 3.3 de la cual se deduce que, cuando el actuador esté a velocidad angular constante ($\ddot{\theta} = 0$), la corriente consumida será proporcional a dicho coeficiente multiplicado por la velocidad.

Para tal propósito se aplicó voltaje a cada actuador y se midió la corriente consumida utilizando un multímetro *Fluke 179*; se determinó la velocidad del rotor a partir de la lectura de la frecuencia a la que entrega pulsos el respectivo *encoder* acoplado, la misma fue medida en un osciloscopio y convertida a *rad/s*; la constante de torque fue tomada de la hoja de datos de los actuadores (Ver Cuadro 3.12).

En el caso particular de la articulación 2, se realizó el ensayo mientras esta se encontraba ensamblada a la articulación 1 (Figura 3.2). Se le dio un voltaje de 30 voltios, se midió la corriente

en la posición de máximo torque y, dado que el *encoder* no se encontraba funcionando, se midió la velocidad angular con un tacómetro por lo que reemplazando los valores medidos en la Ecuación 3.3 se obtuvo la Ecuación 3.7, donde m_{2load} es el peso de la articulación M2; g , la gravedad; R , el factor de reducción y c_m , el centro de masa, considerado en la mitad de la articulación (Ver Anexo 2).

Por otro lado, dado que el actuador de M3 no se encontró disponible para las pruebas y a que este es el mismo modelo que el del actuador de M2, se consideró el mismo coeficiente de rozamiento viscoso; no obstante, este valor sí difiere del valor del actuador de M2, pues los reductores acoplados son diferentes.

$$J2 \times 0 + B \times \dot{\theta} + \frac{m_{2load} * g * c_m}{R} = Kt \times i \quad (3.7)$$

A partir de los ensayos se obtuvieron los valores mostrados en el Cuadro 3.13.

Cuadro 3.13: Cálculo de rozamientos viscosos

Actuador	Velocidad (rad/s)	Corriente (A)	Rozamiento viscoso (Nm.s/rad)
M1	118.66	0.75	8.90×10^{-4}
M2	41.86	1.23	6.54×10^{-3}
M3	41.86	1.23	6.54×10^{-3}
M4	45.10	0.99	6.80×10^{-4}
M5	99.43	0.38	0.091×10^{-3}

Fuente: Elaboración propia

Momentos de inercia referenciados a los actuadores:

Se analizó el momento de inercia de todo el sistema asumiendo el caso en el que la distancia es máxima; es decir, cuando el sistema se encuentra en línea recta.

Se sabe que el momento de inercia de rige bajo la Ecuación 3.8, donde Ja es el momento de inercia en el eje de giro; M_i , las masas manipuladas por la articulación; r_i , la distancia del eje de giro a la masa M_i y n la cantidad de masas manipuladas:

$$Ja = \sum_{i=1}^n M_i \times r_i^2 \quad (3.8)$$

Dado que se trabaja con cuerpos rígidos, para aproximar el análisis, se asumió que la masa de cada elemento está concentrada en su centro de gravedad; por tanto el momento producido por la rotación es dependiente de la masa del cuerpo y la distancia de su centro de gravedad al eje de giro del actuador; así como también, es dependiente de la masa de los actuadores siguientes -asumiéndolos puntuales- y de la distancia de estos al eje de giro (Ver Cuadro 3.2).

Dicho esto, se puede generalizar -exceptuando M5 y M1, pues el eje de giro pasa por el centro de masa de su carga- para el sistema la Ecuación 3.9; de donde m_{pload} es la masa manipulada por

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

el actuador p para el caso en el que el sistema está desarticulado; m_{p+1} , la masa del actuador $p+1$; m_{load} , la masa de la carga final que manipula el robot que puede ser como máximo 1.0Kg; r_{pcm} , la distancia al centro de la masa manipulada por el actuador p ; L_{load} , la distancia a la carga m_{load} que manipula el robot y q , el índice que representa el último actuador manipulado por el actuador p en análisis.

$$J_{pplanta} = m_{pload} * r_{pcm}^2 + m_{p+1} * L_{p+1}^2 + \dots + m_{pload} * r_{qcm}^2 + m_{q+1} * L_q^2 + m_{load} * L_{load}^2 \quad (3.9)$$

A partir de la Ecuación 3.9 se realiza el análisis para el momento de inercia de cada uno de los cinco actuadores utilizados.

Para M4:

$$J_{4planta} = m_{5load} * (L5/2 + L4)^2 + m_5 * L4^2 + m_{4load} * (L4/2)^2 + m_{load} * (L5 + L4)^2 \quad (3.10)$$

Para M3:

$$J_{3planta} = m_{5load} * (L5/2 + L4 + L3)^2 + m_5 * (L4 + L3)^2 + m_{4load} * (L4/2 + L3)^2 + m_4 * L3^2 + m_{3load} * (L3/2)^2 + m_{load} * (L5 + L4 + L3)^2 \quad (3.11)$$

Para M2:

$$J_{2planta} = m_{5load} * (L5/2 + L4 + L3 + L2)^2 + m_5 * (L4 + L3 + L2)^2 + m_{4load} * (L4/2 + L3 + L2)^2 + m_4 * (L3 + L2)^2 + m_{3load} * (L3/2 + L2)^2 + m_3 * L2^2 + m_{2load} * (L2/2)^2 + m_{load} * (L5 + L4 + L3 + L2)^2 \quad (3.12)$$

Para el cálculo del momento de inercia sobre los actuadores M1 y M5, se consideró que dichas articulaciones poseen una geometría cilíndrica hueca y homogénea, para la cual se tiene un momento de inercia de fórmula conocida (Ecuación 3.13); donde m es la masa del cilindro; r_1 , el radio menor y r_2 el radio mayor (Ver Anexo 17 y 15).

$$J = \frac{1}{2}m * (r_1^2 + r_2^2) \quad (3.13)$$

Por tanto, el momento de inercia en M1 sería la suma del momento producido por las otras

articulaciones además del momento producido por la misma articulación asumida cilíndrica de radio r . Por tanto, se tiene la Ecuación 3.14.

$$\begin{aligned}
 J_{1planta} = & m_{5load} * (L5/2 + L4 + L3 + L2)^2 + m_5 * (L4 + L3 + L2)^2 \\
 & + m_{4load} * (L4/2 + L3 + L2)^2 + m_4 * (L3 + L2)^2 \\
 & + m_{3load} * (L3/2 + L2)^2 + m_3 * L2^2 \\
 & + m_{2load} * (L2/2)^2 + m_{load} * (L5 + L4 + L3 + L2)^2 \\
 & + \frac{1}{2} m_{1load} * (r_{11}^2 + r_{12}^2)
 \end{aligned}
 \tag{3.14}$$

De forma análoga a M1, para el actuador de M5 el momento de inercia es el producido por su propia articulación -asumida cilíndrica hueca- y la carga final del robot manipulador; la Ecuación 3.15.

$$J_{5planta} = \frac{1}{2} m_{5load} * (r_{51}^2 + r_{52}^2) + m_{load} * L5^2
 \tag{3.15}$$

Se consideró que el centro de masa de cada estructura que une articulaciones se encuentra a la mitad de la misma; no obstante, esto no es correcto pues el peso de los reductores tiene como efecto que dicho centro de masa se encuentre más cerca de su actuador correspondiente lo cual reduce el momento de inercia. Por otro lado, si el momento de inercia es menor al calculado, entonces el polo que este produce es más lejano al eje imaginario que el polo que se obtiene a partir del modelo hallado; y en consecuencia -teniendo en cuenta solo este criterio- la planta real sería más estable que la modelada. En el Cuadro 3.14 se muestran los valores utilizados para el cálculo de los momento de inercia de todas las articulaciones (J_{planta}) (Ver Anexo 2).

Cuadro 3.14: Datos utilizados para el cálculo del momento de inercia

Articulación	m_p (Kg)	m_{pload} (Kg)	r_{cm} (m)	L_p
1	1.360	7.092	0.0875	0.175
2	1.385	4.901	0.140	0.28
3	1.377	4.434	0.140	0.28
4	0.507	1.730	0.095	0.19
5	0.130	1.973	0.0450	0.09

Fuente: Elaboración propia

Los valores calculados para cada uno de los actuadores se muestran en el Cuadro 3.15 cuales variables halladas son: r_{cm} es la distancia al centro de masa ; J_{planta} ; el momento de inercia calculado previamente sin considerar los factores de reducción; $J_{reductor}$, el momento de inercia producido por los reductores; J_{rotor} , el momento de inercia propio del actuador y J_a , el momento de inercia resultante en el eje del actuador. Es preciso que, para que el sistema esté referenciado al actuador, se consideren los factores de reducción (Ver Cuadro 3.4) sobre la inercia de la planta.

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

Por otro lado, no se consideraron los momentos de inercia de los *encoders* pues estos son muy bajos en comparación al resto de momentos de inercia analizados. En consecuencia, obtenemos la Ecuación 3.16.

$$J_a = \frac{J_{planta}}{R^2} + J_{reductor} + J_{rotor} \quad (3.16)$$

Los valores de $J_{reductor}$ y J_{rotor} fueron tomados de las hojas de datos correspondientes. J_{planta} fue calculado previamente.

Cuadro 3.15: Cálculo del momento de inercia (J_a)

Actuador	$J_{planta} (Kg - m^2)$	$J_{reductor} (Kg - m^2)$	$J_{rotor} (Kg - m^2)$	$J_a (Kg - m^2)$
M1	3.9536	0.036×10^{-3}	4.237×10^{-5}	3.529×10^{-4}
M2	3.9130	0.130×10^{-3}	7.061×10^{-5}	3.535×10^{-4}
M3	1.2355	0.036×10^{-3}	7.061×10^{-5}	1.924×10^{-4}
M4	0.2070	0.014×10^{-3}	1.902×10^{-6}	2.816×10^{-5}
M5	0.0106	0.0033×10^{-3}	1.624×10^{-6}	5.957×10^{-6}

Fuente: Elaboración propia

Simplificación del sistema

Dado que al expandir el denominador de la ecuación de transferencia el valor de la inductancia que multiplicado por la inercia y el coeficiente de rozamiento viscoso generan valores muy bajos en comparación al resto de componentes; por tanto, es factible despreñar la inductancia [21] [24]; en consecuencia, se obtiene el modelo simplificado mostrado en la ecuación 3.17.

$$\frac{\theta(s)}{v(s)} = \frac{Kt}{[Ra * Ja * s + Ra * B + Kt * Ke] * s} \quad (3.17)$$

Diseño del controlador PID

El controlador debe lograr un sobreimpulso de 0% para evitar cambiar de sentido el actuador continuamente; además, debe lograr el menor tiempo de establecimiento posible.

Para obtener los parámetros se sintonizó mediante ajuste manual a partir de la sintonización automática propia del software utilizado: *Simulink* de *Matlab*; el permite observar la respuesta de la planta ante diferentes entradas para así determinar si el controlador que se está sintonizando es adecuado para el sistema.

Dado que el sistema ya posee un integrador en su ecuación de transferencia, se consideró solo un controlador PD para cada articulación de control clásico [19] (Ecuación 3.18).

$$D(s) = K_P + K_D \times s \quad (3.18)$$

El análisis de cada articulación se realizó a partir del diagrama de bloques de la Figura 3.20, considerando los factores de reducción (Cuadro 3.4), los valores de inercia referenciados al actuador (Cuadro 3.15), la resistencia de armadura y constante de torque-velocidad (Cuadro 3.12), los coeficientes de rozamiento viscoso 3.13 y la cantidad de pulsos por vuelta de los *encoders* (Cuadro 3.6).

Para la simulación no se tomó en cuenta la zona muerta del actuador, pues dicha consideración es parte del escalamiento *PWM*; no obstante, sí se consideró una saturación máxima de +15 y -15 voltios para la salida del controlador, mencionada previamente. Las simulaciones se realizaron para un punto de referencia de π rad (180°). En las gráficas halladas se muestra también las características de la curva encontradas utilizando la función *stepinfo*; se debe acotar que dicha función se manipuló simulando con un *steptime* de 0 segundos; sin embargo, para mayor claridad las gráficas se muestran con un *steptime* de 0.5 segundos.

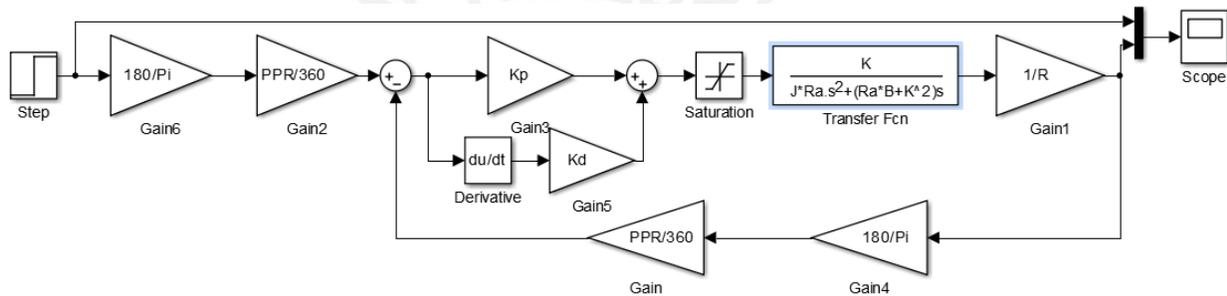


Figura 3.20: Diagrama de bloques de la simulación del sistema de control de la articulación

Para M1:

El modelo matemático resultante es:

$$\frac{\theta(s)}{v(s)} = \frac{0,141}{[0,008802 * s + 0,04209] * s} \tag{3.19}$$

El controlador optimizado es el siguiente:

$$D(s) = 1,0 + 0,2 * s \tag{3.20}$$

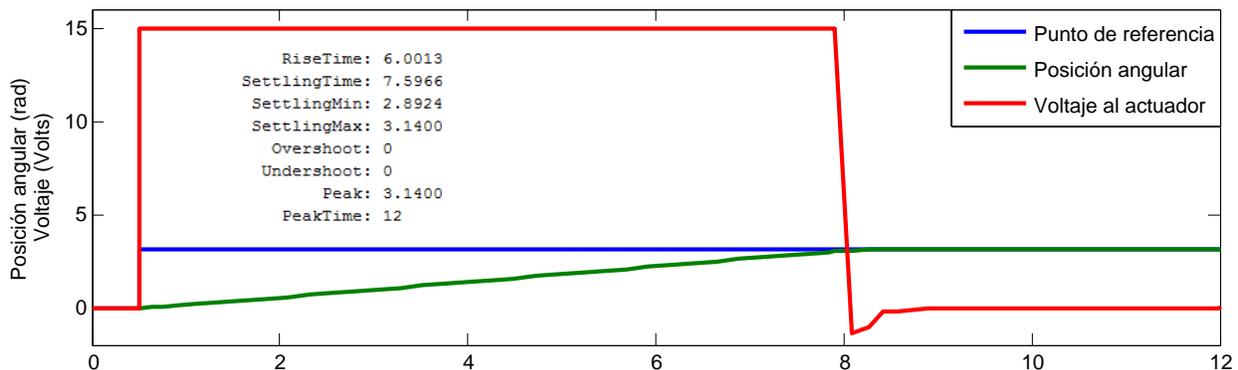


Figura 3.21: Respuesta a un pulso. Articulación 1

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

En la simulación se aprecia un error de estado estable de 0, sobreimpulso de 0% y un tiempo de establecimiento de 7.59 segundos.

Para M2:

El modelo matemático resultante es:

$$\frac{\theta(s)}{v(s)} = \frac{0,257}{[0,007372 * s + 0,2027] * s} \tag{3.21}$$

El controlador hallado es el siguiente:

$$D(s) = 1,3 + 0,1 * s \tag{3.22}$$

En la simulación se aprecia un error de estado estable de 0, sobreimpulso cercano a cero y un tiempo de establecimiento de aproximadamente 25.92 segundos.

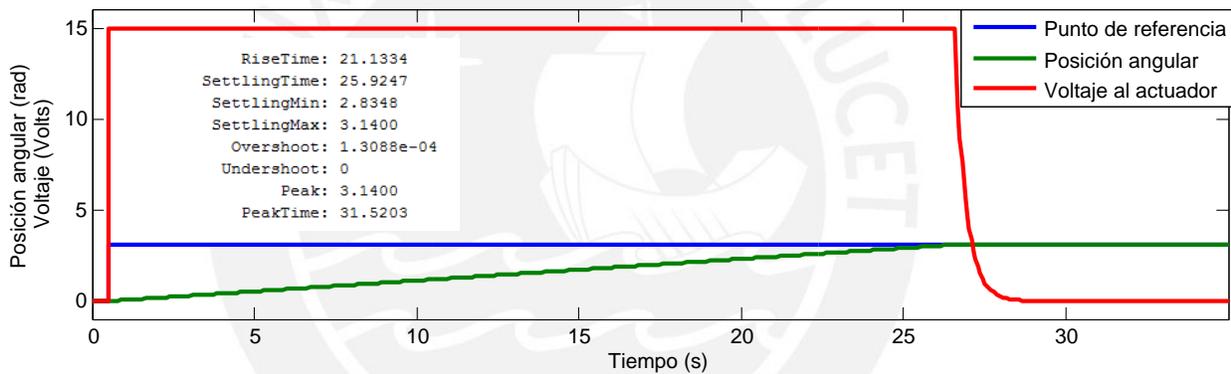


Figura 3.22: Respuesta a un pulso. Articulación 2

Para M3:

El modelo matemático resultante es:

$$\frac{\theta(s)}{v(s)} = \frac{0,257}{[0,004016 * s + 0,2029] * s} \tag{3.23}$$

El controlador hallado es el siguiente:

$$D(s) = 1,2 + 0,1 * s \tag{3.24}$$

En la simulación se aprecia un error de estado estable de 0, sobreimpulso cercano a cero y un tiempo de establecimiento de aproximadamente 19.45 segundos.

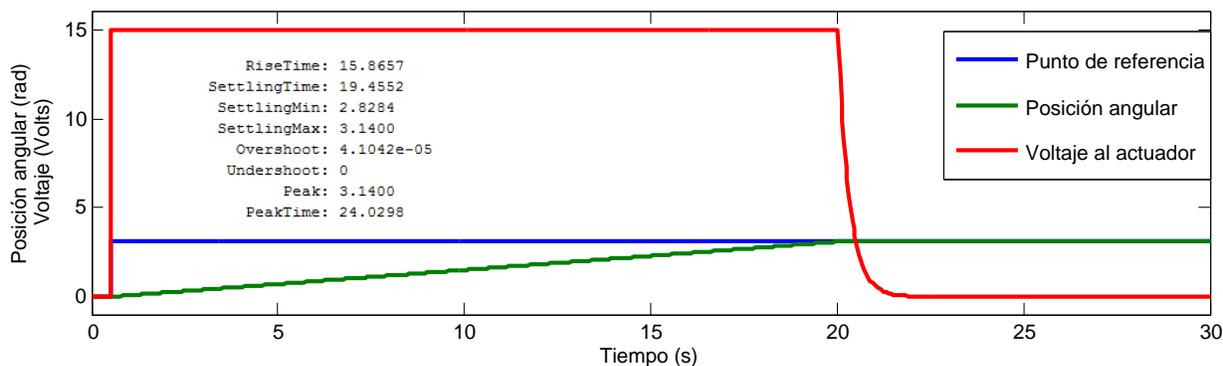


Figura 3.23: Respuesta a un pulso. Articulación 3

Para M4:

El modelo matemático resultante es:

$$\frac{\theta(s)}{v(s)} = \frac{0,031}{[0,0006173 * s + 0,003597] * s} \tag{3.25}$$

El controlador hallado es el siguiente:

$$D(s) = 0,7 + 0,15 * s \tag{3.26}$$

En la simulación se aprecia un error de estado estable de 0, sobreimpulso de 0% y un tiempo de establecimiento de aproximadamente 3.5 segundos.

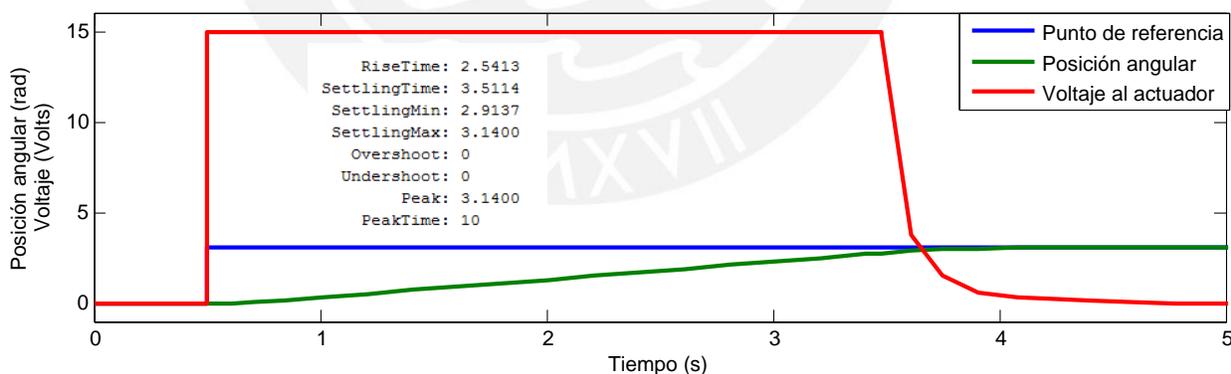


Figura 3.24: Respuesta a un pulso. Articulación 4

Para M5:

El modelo matemático resultante es:

$$\frac{\theta(s)}{v(s)} = \frac{0,024}{[0,00002463 * s + 0,000941] * s} \tag{3.27}$$

En este caso se encontró que un controlador P es más adecuado que uno PD , esto se debe a que el polo que posee la planta se encuentra más alejado del eje imaginario en comparación a los

otros actuadores; por tanto, la planta es más estable.

$$D(s) = 0,5 \tag{3.28}$$

En la simulación se aprecia un error de estado estable de 0, sobreimpulso de 0% y un tiempo de establecimiento de aproximadamente 0.87 segundos.

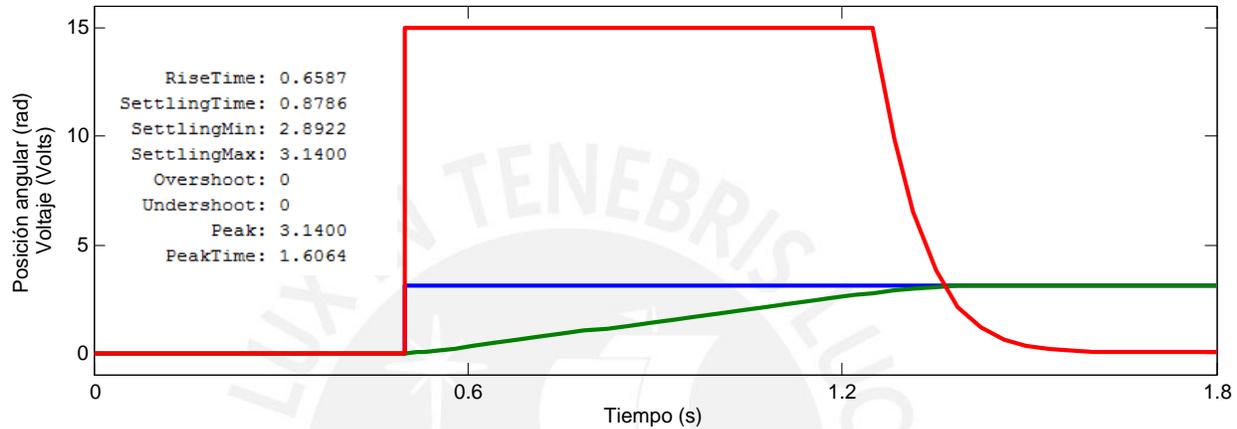


Figura 3.25: Respuesta a un pulso. Articulación 5

Implementación del algoritmo

Transformando el controlador PD clásico continuo (Ecuación 3.18) al tiempo discreto mediante la aproximación de Euler se obtiene la Ecuación 3.29 [25]:

$$D(z) = K_P + K_D * \frac{z - 1}{Tz} \tag{3.29}$$

Aplicando la transformada inversa de Z para la entrada error recibido ($e[k]$) y considerando periodo de muestreo T , se obtiene la ecuación de diferencias (Ecuación 3.30) que describe el algoritmo utilizado en cada controlador.

$$Gc[k] = K_P * e[k] + K_D * \frac{1}{T} * (e[k] - e[k - 1]) \tag{3.30}$$

En la Figura 3.26 se muestra el diagrama de flujo del algoritmo de control implementado en los controladores esclavos, obtenido a partir de las ecuaciones de diferencias respectivas.

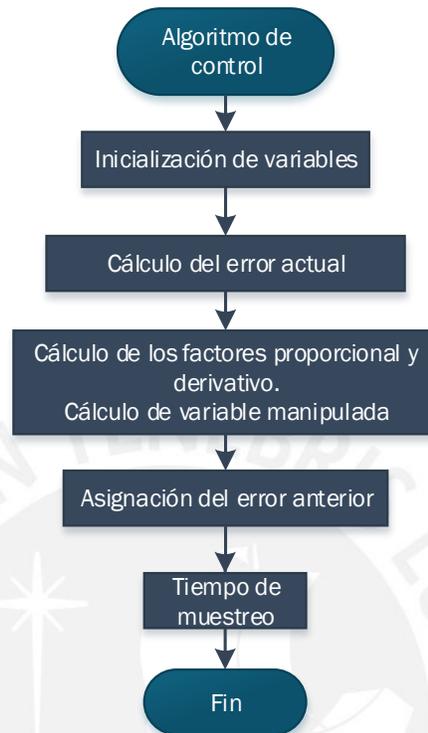


Figura 3.26: Diagrama de flujo del algoritmo de control

Primero se inicializan las variables a utilizar; para esto se asignó el tipo *float* para las variables *error_actual*, *error_anterior*, *factor_proporcional* y *factor_derivativo*. Luego se calcula el valor de la variable manipulada y se asigna a la variable *error_anterior* el valor del error actual calculado. Posteriormente se realiza el escalamiento necesario para la onda *PWM* y se determina el sentido de giro, estos valores son enviados al excitador a través de los pines asignados previamente (Figura 3.7). Finalmente se realiza un retraso T que representa el periodo de muestreo.

Sistema estable

En todas las simulaciones se encontró un rizado en el estado estacionario (Figura 3.27), siendo el más crítico de 0.02 %; para reducir esta oscilación se considera que el sistema se encuentra estable cuando el error medido es de $\pm 0.5\%$ durante un periodo de 500 veces el periodo de muestreo. En dicho estado se deja de responder al algoritmo de control y se envía al excitador un nivel bajo de voltaje. No obstante, para dicho instante de estabilidad, el sistema aun se encuentra dentro del algoritmo de movimiento; por tanto, si el error fuera nuevamente mayor a $\pm 0.5\%$ se atenderá al algoritmo de control. Esta condición define en última instancia la precisión del sistema.

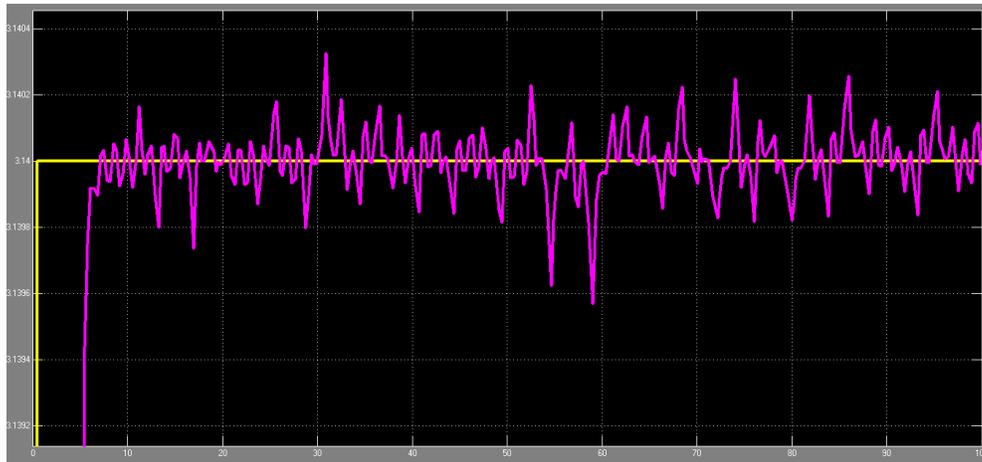


Figura 3.27: Rizado en estado estacionario

Periodo de muestreo

El periodo de muestreo (T) se genera mediante la función `_delay_us`, propia de la librería *AVR Libc* que es incluida en el entorno de programación utilizado. Dicha función retrasa la ejecución del programa durante un tiempo solicitado en microsegundos [3]; es preciso mencionar que dicha función no afecta la normal ejecución de las subrutinas de interrupción. Para elegir el menor tiempo de muestreo se utilizó el criterio representado por la Ecuación 3.31 [18], donde t_r es el tiempo de subida de la planta en lazo abierto.

$$T \leq \frac{t_r}{10} \quad (3.31)$$

El caso más crítico es el de la articulación 5, la cual requeriría un periodo de muestreo menor a 65ms, por lo que se eligió un periodo de muestreo de 10ms y se consideró el mismo para los controladores de todas las articulaciones.

3.4.3.2. Lectura del encoder

Se guardó la cuenta de pulsos en una variable modificada por la subrutina de interrupción externa (*ISR_INT0*); dicha subrutina se activa en cada flanco de subida del canal A del *encoder*. La variable se incrementa si es que el sentido de giro del actuador es igual al sentido de giro hallado por el software y se decrementa en el caso contrario. Se puede saber el sentido de giro mediante la lectura del estado del pin asignado al canal B, si dicho pin está en alta durante el inicio de la subrutina de interrupción, el sentido es horario y es antihorario cuando el pin esté en baja.

3.4.3.3. Escalamiento PWM

Dado que el valor numérico de la salida del controlador (u) representa un valor de voltaje, es necesario realizar un escalamiento para que este pueda manipular efectivamente el ciclo de trabajo

CAPÍTULO 3. DISEÑO E IMPLEMENTACIÓN DE LOS CONTROLES ELECTRÓNICOS E INTERFAZ DE USUARIO

de la onda *PWM* configurada; y, de esta forma, manipular el voltaje promedio que llega finalmente a los actuadores. En este escalamiento se considera la zona muerta de los actuadores, encontradas mediante pruebas realizadas.

La entrada de la etapa de escalamiento -o salida del controlador- va desde cero voltios hasta el valor de saturación (15 Voltios) y la salida es el valor del registro *OCR1A* que modifica el ciclo de trabajo de la onda *PWM*. La salida tiene un rango que va desde el valor en el cual se inicia el movimiento para cada actuador (Lim_p) hasta el valor de saturación correspondiente al módulo *PWM* (255).

Como es una relación lineal se deduce la Ecuación 3.32. Los valores de Lim_p -encontrados mediante ensayos- se muestran en el Cuadro 3.16.

$$OCR1A = \frac{255 - Lim_p}{15} \times u + Lim_p \quad (3.32)$$

Cuadro 3.16: Límite inferior de saturación de voltaje

Actuador	Lim_p
M1	45
M2	150
M3	140
M4	60
M5	35

Fuente: Elaboración propia

Capítulo 4

Ensayos

En el presente capítulo se detallan las pruebas realizadas al sistema implementado, esto comprende pruebas a la interfaz de usuario, microcontrolador maestro y microcontroladores esclavos. Es preciso resaltar que al momento de las pruebas solo se encontraban la primera y la segunda articulación ensambladas y que el *encoder* acoplado a M2 no estaba funcionando. Las pruebas realizadas fueron las que se mencionan a continuación.

- A la interfaz de usuario: búsqueda, conexión y desconexión del puerto serial y generación y envío de la trama.
- Al microcontrolador maestro: recepción de la trama serial, detección de estado ocupado y envío de trama I^2C a los cinco microcontroladores esclavos.
- A los microcontroladores esclavos: recepción de trama I^2C , respuesta ante fin de carrera, correcto cálculo del sentido de giro y respuesta en lazo cerrado.

4.1. Interfaz de usuario

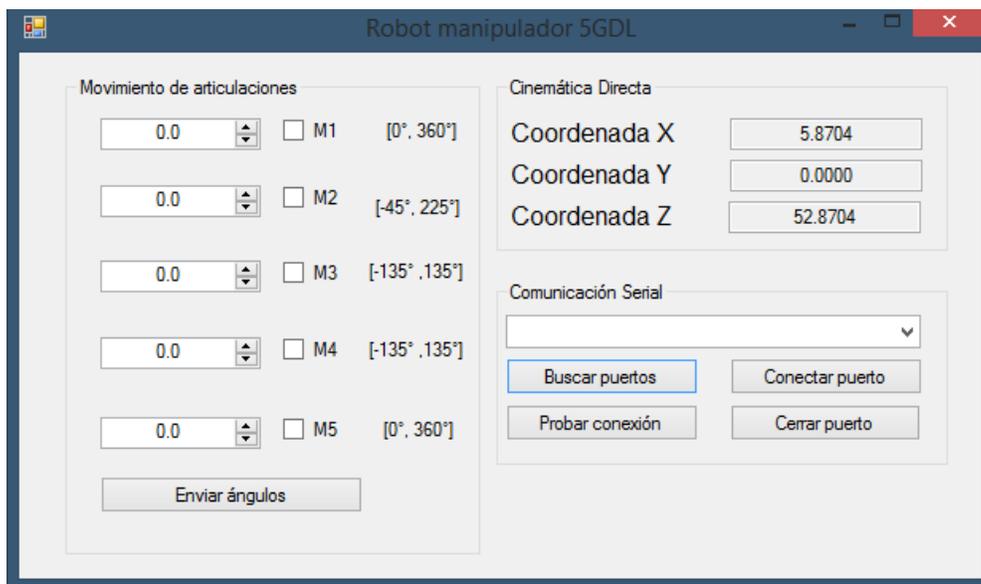


Figura 4.1: Interfaz de usuario

La Figura 4.1 muestra la interfaz de usuario implementada. El procedimiento de uso inicia con realizar la conexión serial, para esto se debe buscar el puerto, seleccionarlo y abrir la conexión; en caso de error o éxito se muestran los mensajes correspondientes. Adicionalmente, se posee un botón de prueba para conocer el estado del maestro. Los mensajes mostrados en la Figura 4.2 son algunos de los resultados de las pruebas realizadas a la comunicación serial de la interfaz.

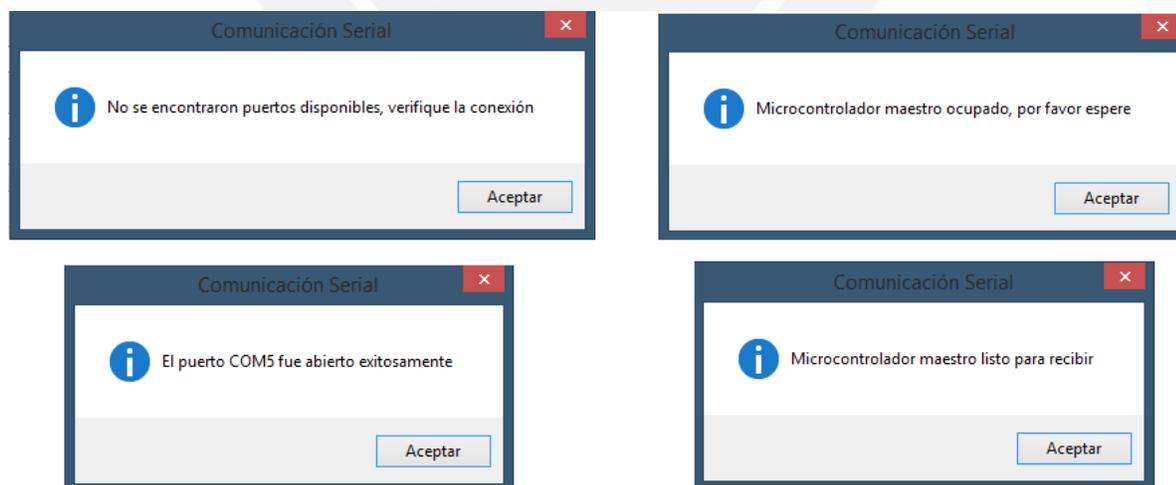


Figura 4.2: Interfaz de usuario

Luego, se seleccionan los actuadores que se desea mover y se ingresa el ángulo requerido; estos datos son enviados vía serial al maestro en el caso de que este se encontrara disponible para recibir.

En la Figura 4.3 se observa la lectura en osciloscopio de la trama enviada para cinco esclavos,

CAPÍTULO 4. ENSAYOS

se comprueba que esta llega correctamente en un tiempo de 33ms. Se puede distinguir la consulta que realiza la interfaz (1), la respuesta del microcontrolador maestro (2) y la trama de ángulos enviada por la interfaz (3).

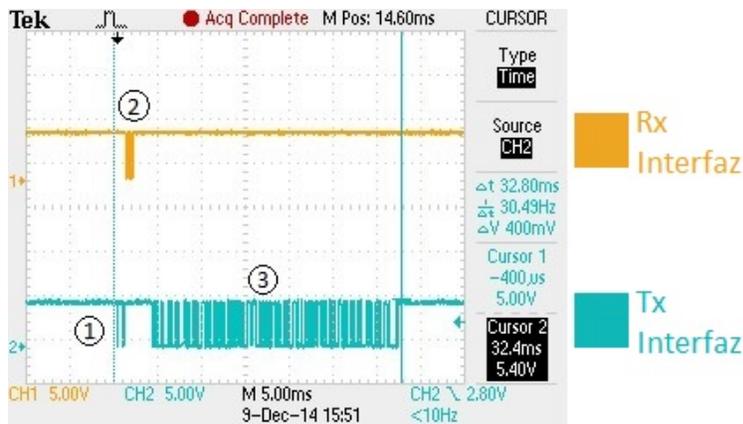


Figura 4.3: Trama serial visualizada

4.2. Microcontrolador maestro

En la Figura 4.4 se muestra la trama por I^2C enviada desde el maestro a dos esclavos conectados (SDA) en línea celeste y el reloj utilizado en el protocolo (SCL) en línea naranja. La trama tiene una duración de 14ms e implica leer el estado de ocupado del buffer del esclavo y escribir 4 bytes con el ángulo solicitado.

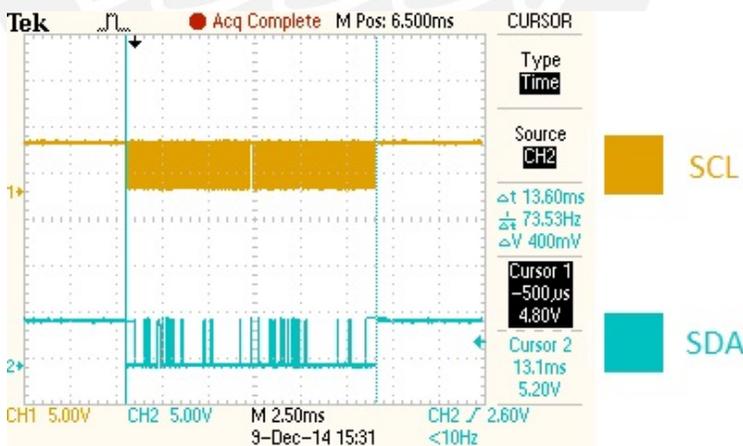


Figura 4.4: Trama I^2C para dos esclavos

4.3. Microcontroladores esclavos

Las pruebas realizadas en los esclavos fueron las siguientes: Recepción de comunicación, respuesta ante fin de carrera, correcto cálculo del sentido de giro y también se obtuvo la curva de respuesta para 180° .

CAPÍTULO 4. ENSAYOS

En la prueba de fin de carrera se comprobó que el sistema se detiene y guarda efectivamente la nueva posición ante el accionamiento del sensor asigando. De igual forma, el cálculo de sentido más corto y de cantidad de pulsos a controlar se realiza de forma efectiva. Estos datos fueron comprobados a partir de la lectura del buffer I^2C que cada esclavo posee. Se implementó un programa en una placa *Arduino Mega2560* que leyó dicho buffer y envió vía serial a un computador los datos obtenidos.

En la Figura 4.5 se muestra la posición final de la articulación 1 para los puntos de referencia de 0° , 90° , 180° y 270° .

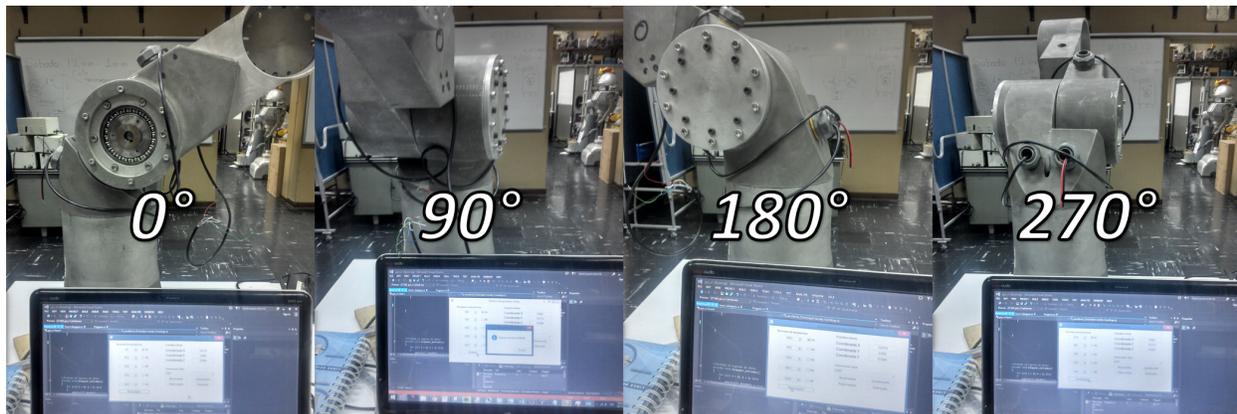


Figura 4.5: Posición angular de la articulación M1

Para obtener la curva de respuesta se implementó también un programa en la placa *Arduino Mega2560*, mencionada previamente. A dicha tarjeta se le conectó a sus pines los canales A y B del *encoder* y se realizó la lectura por interrupción de la misma forma implementada en los microcontroladores esclavos; dicha información fue enviada cada 4ms vía serial a un computador. Para mayor entendimiento, los datos recibidos fueron escalados a su equivalente en grados sexagesimales y fueron graficados utilizando la función *plot* de *Matlab*; con la función *stepinfo* se obtuvieron las características de la curva.

En la Figura 4.6 se observa la respuesta de la articulación 1 para un punto de referencia de 180° .

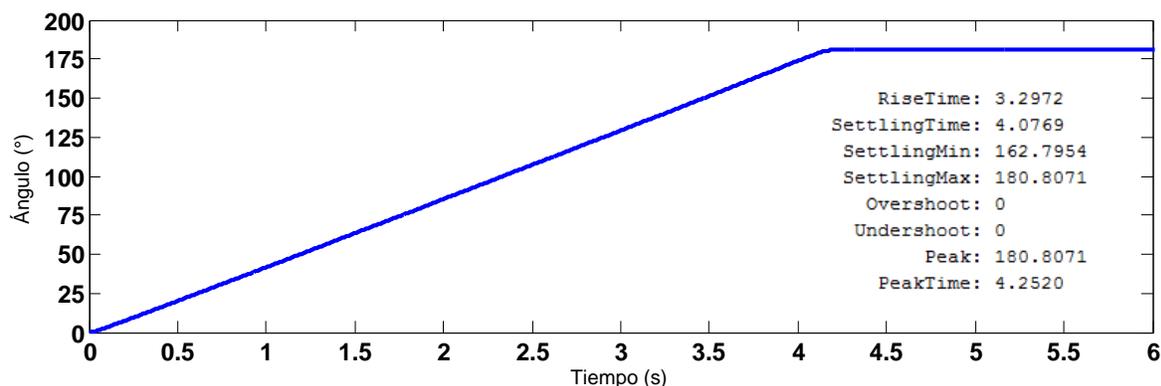


Figura 4.6: Respuesta de control de posición - Articulación 1

CAPÍTULO 4. ENSAYOS

De la curva se obtiene un error de 0.807° equivalente a 0.4%, un sobreimpulso de 0% y un tiempo de establecimiento de 4.07 segundos.

Adicionalmente, derivando la curva de posición obtenida previamente (Figura 4.6), se graficó la curva de velocidad de la articulación 1 mostrada en la Figura 4.7. Para dicho propósito se utilizó la función *diff* y *plot* de *Matlab*.

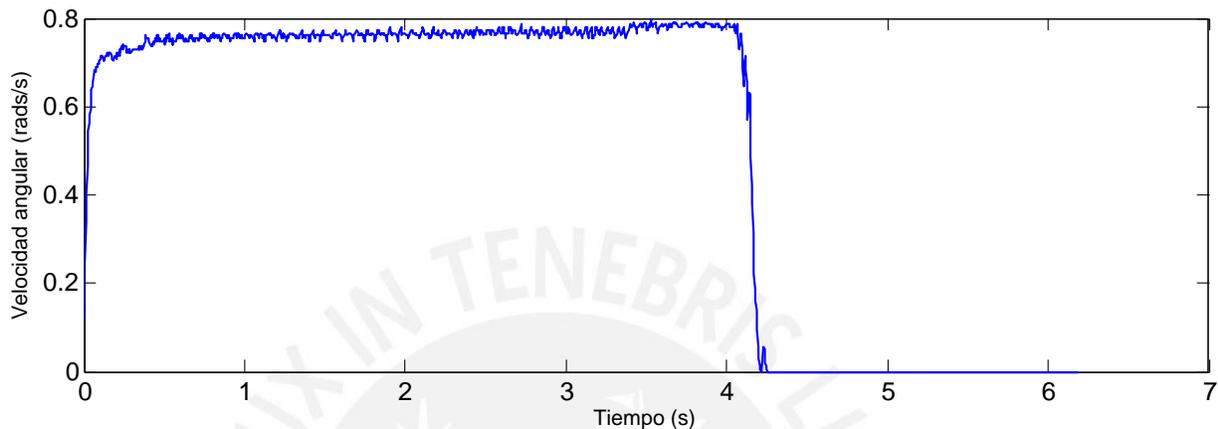


Figura 4.7: Respuesta en velocidad de M1

Se observa que la planta es más rápida que las simulaciones realizadas; esto se debe, principalmente, a que algunos de los valores de los coeficientes del modelo matemático obtenido dependen en gran parte de aproximaciones asumidas. Un caso particular es el del coeficiente de rozamiento viscoso (*B*) pues este es usualmente despreciado; no obstante, en los ensayos realizadas, se obtuvo un valor muy elevado pues no se consideraron torques de perturbación que pudieran existir como pueden ser torque de fricción de cinética, fricción de Coulomb, fuerza centrípeta y fuerza coriolis [11] [20]. Para tal caso, despreciando el coeficiente de rozamiento viscoso, se realiza la simulación para la articulación 1, mostrada en la Figura 4.8. Esto afecta especialmente a las articulaciones 1 y 2, pues al momento de los ensayos estas se encontraban ensambladas; a diferencia de las articulaciones 4 y 5, que se encontraban acopladas solo a su respectivo reductor.

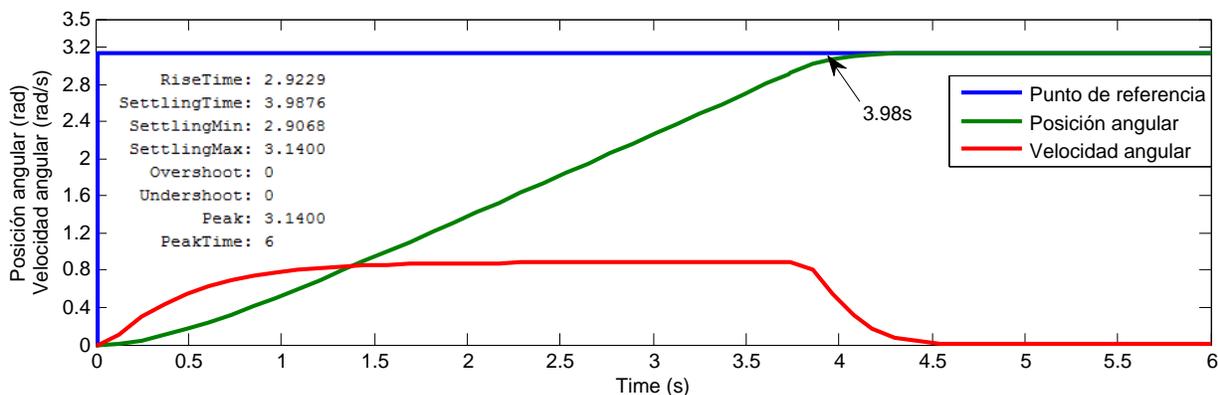


Figura 4.8: Respuesta de la simulación

De la gráfica (Figura 4.8) se observa un tiempo de establecimiento de 3.98 segundos obtenido

CAPÍTULO 4. ENSAYOS

con la función *stepinfo*, este valor es mucho más cercano al valor real obtenido; por lo que es posible afirmar que la eliminación del coeficiente de rozamiento viscoso es factible.

Dado que el microcontrolador utilizado opera de manera secuencial, el tiempo de muestreo depende directamente de la duración del algoritmo de control, por tanto este fue calibrado a partir de un ensayo realizado. Para esto, se implementó un programa en la placa *Arduino* mencionada previamente, el cual midió la duración de pulsos de un pin -de un microcontrolador esclavo- que es puesto en alta al entrar a la subrutina de control y en baja justo antes de entrar a un retardo de $25\mu s$ y salir de la subrutina.

En la Figura 4.9 se muestra la variación de la duración de la subrutina de control para un punto de referencia de 180° ; en la misma se aprecia una gran variación en su duración, la cual se debe a la constante multiplicación y división de números de punto flotante durante la subrutina en cuestión, lo cual implica estructuras iterativas que pueden aumentar considerablemente el tiempo de procesamiento. Para calibrar el tiempo de muestreo se tomó la situación más crítica, que es cuando la duración de la subrutina de control es la más elevada, siendo esta aproximadamente $400\mu s$. Entonces, para llegar al tiempo de muestreo elegido de $10ms$, se calibró el retardo dado a $9.6ms$, cumpliendo de esta manera con el valor elegido.

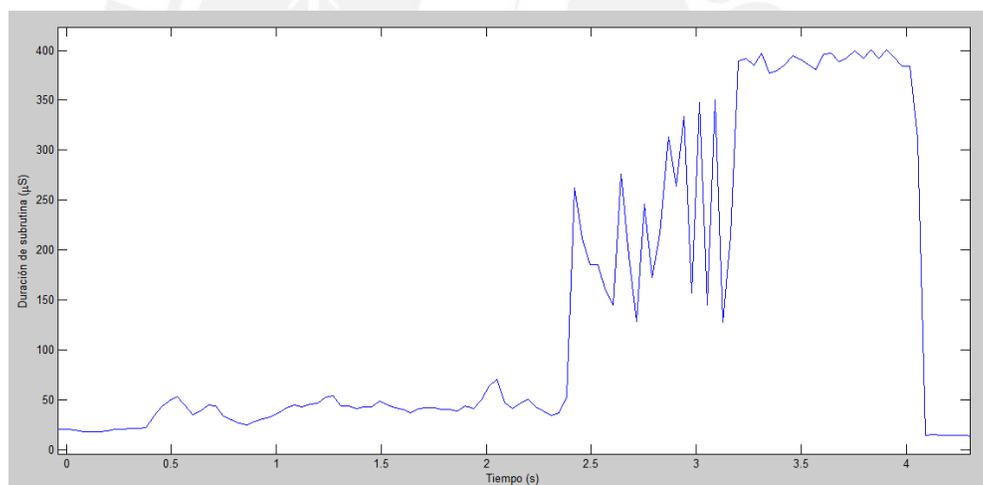


Figura 4.9: Duración de la subrutina de control

Conclusiones

Del diseño e implementación mostrados en el presente trabajo de tesis se puede concluir que:

La interfaz de usuario respondió de forma eficaz ante las pruebas realizadas, manipulando la comunicación serial del computador, generando la trama solicitada y enviándola vía serial al microcontrolador maestro. Se comprobó que la velocidad de 9600bps era suficiente para la aplicación, pues el tiempo total de trama medido (33ms) fue imperceptible al momento de hacer las pruebas.

Se comprobó también que el microcontrolador maestro recibe correctamente los datos ingresados y envía la respectiva trama a cada esclavo correspondiente.

Se obtuvo la curva de posición que constata que los microcontroladores reciben el ángulo solicitado y efectúan de forma correcta el algoritmo de control implementado. Ante la comparación con la simulación del modelo matemático utilizado de la articulación 1, se encontró que el modelo real es mucho más rápido que el esperado; se comprobó que se obtiene una mucho mejor aproximación al modelo real despreciando el coeficiente de rozamiento viscoso.

Finalmente, se concluye que, a partir de la programación desarrollada de la interfaz y los microcontroladores -bajo el esquema Maestro-Eslavos-, se llegó de forma eficaz a la posición angular de la articulación 1, solicitada por el usuario. Se comprobó además que la información llega a los esclavos de forma exitosa; y, dado que la programación de cada esclavo es la misma, es posible afirmar que se cumplió con el objetivo principal. No obstante, es necesario realizar pruebas con las demás articulaciones conectadas para contrastarlas con las simulaciones realizadas pues, al momento de la finalización del presente trabajo de tesis, estas aún no se encontraban ensambladas.

Recomendaciones

Se debe considerar la adquisición de un excitador que pueda operar con mayor voltaje, pues los actuadores de los que se dispone se encuentran muy limitados por el excitador actual; esto permitiría lograr un mejor tiempo de establecimiento.

Para agregar funciones (p.e.: adicionar control de velocidad) es necesario aumentar la velocidad de trabajo de los microcontroladores, para esto se puede agregar un oscilador externo para trabajar el microcontrolador a 16Mhz o, en su defecto, cambiar de unidad de procesamiento, pudiéndose considerar microcontroladores con unidad de punto flotante integrada, de varios núcleos o implementar el sistema en un *FPGA*.

Dado que el protocolo de comunicación I^2C es ideado para comunicación entre periféricos en una misma tarjeta [7] se recomienda la utilización de otro protocolo de comunicación, como podría ser *RS485* o *busCAN* que permiten comunicaciones a mayor distancia y mayor velocidad. A partir de esto se podría considerar la implementación de los controladores en el computador personal para lo cual los esclavos se limitarían a recibir solo las variables de control en todo momento y no el ángulo a girar; esto permitiría realizar un control del sistema más complejo y preciso; no obstante se requería un bus de datos más rápido y estable.

Bibliografía

- [1] *Diccionario de la Real Academia Española*. Libr. de Parmantier, 1826. [Online]. Available: <http://books.google.com.pe/books?id=OanYapiW-OIC>
- [2] “1961: Installation of the first industrial robot,” Noviembre 2014. [Online]. Available: <http://world-information.org/wio/infostructure/100437611663/100438659325>
- [3] “Avr-libc: función delay,” Noviembre 2014. [Online]. Available: http://www.nongnu.org/avr-libc/user-manual/group_util_delay.html
- [4] “Dc motor speed: System modeling,” Noviembre 2014. [Online]. Available: <http://ctms.engin.umich.edu/CTMS/index.php?example=MotorSpeed§ion=SystemModeling>
- [5] “Dual vnh2sp30 motor driver carrier md03a,” Noviembre 2014. [Online]. Available: <http://www.pololu.com/product/708/specs>
- [6] “Epson g20,” Noviembre 2014. [Online]. Available: <http://robots.epson.com/product-detail/5>
- [7] “I2c bus, what is that?” Noviembre 2014. [Online]. Available: <http://www.i2c-bus.org/>
- [8] “Introducción a la robótica,” Noviembre 2014. [Online]. Available: <http://slideplayer.es/slide/92174/>
- [9] “Lo nuevo en visual basic y visual c,” Noviembre 2014. [Online]. Available: [http://msdn.microsoft.com/es-pe/library/aa984213\(v=vs.71\).aspx](http://msdn.microsoft.com/es-pe/library/aa984213(v=vs.71).aspx)
- [10] “Manipulador neumático cartesiano avm,” Noviembre 2014. [Online]. Available: <http://www.directindustry.es/prod/avm-automation/manipuladores-neumaticos-cartesianos-8908-22568.html>
- [11] “Massachusetts institute of technology, solving the equation of motion,” Noviembre 2014. [Online]. Available: <http://ocw.mit.edu/courses/mechanical-engineering/2-004-systems-modeling-and-control-ii-fall-2007/lecture-notes/lecture02.pdf>
- [12] “North american robotics market sets new records for first nine months of 2014,” Noviembre 2014. [Online]. Available: <http://www.robotics.org/content-detail.cfm/Industrial-Robotics-News/>

BIBLIOGRAFÍA

- North-American-Robotics-Market-Sets-New-Records-for-First-Nine-Months-of-2014/
content_id/5086
- [13] “R19 4-axis robot arm,” Noviembre 2014. [Online]. Available: <http://www.strobotics.com/cylindrical-format-robot.htm>
- [14] “Robot industrial fanuc r2000ia/200r,” Noviembre 2014. [Online]. Available: <http://www.exapro.es/robot-industrial-fanuc-r2000ia200r-p40115124/?counter=20&category=685&page=1#/?>
- [15] “Robotics history timeline - robotics research group - the university of auckland,” Noviembre 2014. [Online]. Available: http://robotics.ece.auckland.ac.nz/index.php?option=com_content&task=view&id=31
- [16] “Robots and robotic devices vocabulary,” Noviembre 2014. [Online]. Available: <https://www.iso.org/obp/ui/#iso:std:iso:8373:ed-2:v1:en>
- [17] “Robots drive systems,” Noviembre 2014. [Online]. Available: <http://www.roboticsbible.com/robot-drive-systems.html>
- [18] K. J. Åström and B. Wittenmark, *Computer-Controlled Systems*. Department of Automatic Control - Lund Institute of Technology, 1997.
- [19] K. J. Astrom and R. M. Murray, *Feedback Systems*. Princeton University Press, 2008.
- [20] A. Barrientos, *Fundamentos de Robótica*. McGraw-Hill, 1997.
- [21] L.-A. Dessaint, B. Hebert, and H. Le-Huy, “An adaptive controller for a smooth positioning system: analysis and simulation,” in *Industrial Electronics Society, 1989. IECON '89., 15th Annual Conference of IEEE*, Nov 1989, pp. 257–262 vol.2.
- [22] D. Foster and A. Harrison, “Experimental investigation of a 5-dof robot arm with sliding mode control,” in *Robotics, Automation and Mechatronics (RAM), 2011 IEEE Conference on*, Sept 2011, pp. 125–130.
- [23] P. Mota, J. Pierre Rognon, and H. Le-Huy, “Digital position servo system: A state variable feedback system,” *Industry Applications, IEEE Transactions on*, vol. IA-20, no. 6, pp. 1473–1481, Nov 1984.
- [24] A. Ollero Baturone, *Robótica; manipuladores y robots móviles*. Marcobombo, 2001.
- [25] C. L. Phillips and H. T. Nagle, *Digital Control System Analysis and Design*. Prentice Hall, Inc., 1995.